



中南大學

CENTRAL SOUTH UNIVERSITY

应用基础实践一（网络+Java）

报 告

姓 名：卜德华

学 号：8208220314

专业班级：大数据 2201

指导教师：黎方正

完成时间：2024 年 6 月 30 日

中南大学计算机学院

2024

目录

一、开发工具及环境说明.....	1
1.1、开发工具：	1
1.2、环境说明：	1
二、网络聊天程序业务分析.....	2
2.1 实验目的	2
2.2 需求设计	2
三、网络聊天程序系统设计.....	4
3.1 聊天室设计流程图	4
3.2 关键问题设计	5
四、网络聊天程序源代码清单.....	8
4.1 客户端创建主窗体并连接服务器	8
4.2 客户端注册、登录	9
4.3 客户端私公聊，收发文件，发送表情，发送窗口震动	11
4.4 客户端历史记录	17
4.5 服务器创建主窗体	17
4.6 服务器私发消息、广播、踢人、禁言及解除禁言	18
五、网络聊天程序运行结果与测试分析.....	22
5.1 注册登录界面	22
5.2 客户端私聊、公聊界面	23
5.3 服务器发送系统消息、踢人、禁言界面	24
5.4 发送文件	25
5.5 客户端历史记录	26
六、爬取并分析网页敏感词程序业务分析.....	27
6.1 实验目的	27
6.2 需求设计	27
6.3 实现思路：	28
七、爬取并分析网页敏感词程序系统设计.....	29
7.1 爬取并分析网页敏感词流程图	29
7.2 爬取并分析网页敏感词代码文件	30
八、爬取并分析网页敏感词源代码清单.....	31
8.1 Start. java	31
8.2 URLfilter. java	32
九、爬取并分析网页敏感词程序运行结果与测试分析.....	66
9.1 运行代码	66
9.2 主界面	66
9.3 点击爬取目标 url.....	67
9.4 点击导入敏感词库	68
9.5 点击高亮显示	69
9.6 点击导入目标 url 库.....	70
9.7 点击开始批量爬取	70
9.8 查看爬取内容和敏感词	72

十、结论与心得.....	73
10.1 网络聊天程序	73
10.2 爬取并分析网页敏感词程序	75

一、开发工具及环境说明

1.1、开发工具：

IntelliJ IDEA

1.2、环境说明：

集成开发环境（IDE）

二、网络聊天程序业务分析

2.1 实验目的

用 Java Socket 编程开发聊天室，该聊天室由服务器端和客户端两部分构成，其主要功能有：

1. 支持多个客户端连接到一个服务器，每个客户端能够输入账号。
2. 可以实现群聊（聊天记录显示在所有客户端界面）。
3. 完成好友列表在各个客户端上显示。
4. 可以实现私人聊天，用户可以选择某个其他用户，单独发送信息。
5. 服务器能够群发系统消息，能够强行让某些用户下线。
6. 客户端的上线下线要求能够在其他客户端上面实时刷新。

2.2 需求设计

1. 多个客户端连接和账号输入：
 - 客户端连接信息可以存储在服务器端的内存中，使用数据结构如列表或映射来管理连接状态。
 - 账号信息可以存储在服务器端的数据库中，包括用户名和密码。可以使用关系型数据库如 MySQL，通过表来存储账号信息。
2. 群聊功能：
 - 聊天记录可以存储在服务器端的数据库中。可以创建一个聊天记录表，包含字段如消息内容、发送者、发送时间等。每当有新的群聊消息时，将消息插入到聊天记录表中，并通过查询将聊天记录显示在所有客户端界面上。
3. 好友列表：
 - 好友列表可以存储在服务器端的数据库中。可以创建一个用户表和好友关系表。用户表包含字段如用户名、唯一标识符等，好友关系表包含字段如用户标识符和好友标识符。通过查询好友关系表，服务器可以获取每个客户端的好友列表，并将其在各个客户端上显示。
4. 私人聊天：
 - 私人聊天消息可以存储在服务器端的数据库中，类似于群聊功能中的聊天记录表。每当有新的私人聊天消息时，将消息插入到私人聊天记录表中，并通过查询将私人聊天记录显示在相应的客户端界面上。
5. 系统消息和用户强制下线：
 - 系统消息可以存储在服务器端的数据库中，类似于群聊功能中的聊天记录表。服务器可以向特定的聊天室或所有在线用户发送系统消息，将消息插入到系

统消息记录表中，并通过查询将系统消息显示在客户端界面上。

- 用户强制下线可以在服务器端的内存中管理。服务器可以维护一个用户状态列表，标记用户的在线状态。当需要强制下线某个用户时，服务器将该用户的状态设置为离线，断开与该用户的连接。

6. 客户端的上线下线实时刷新：

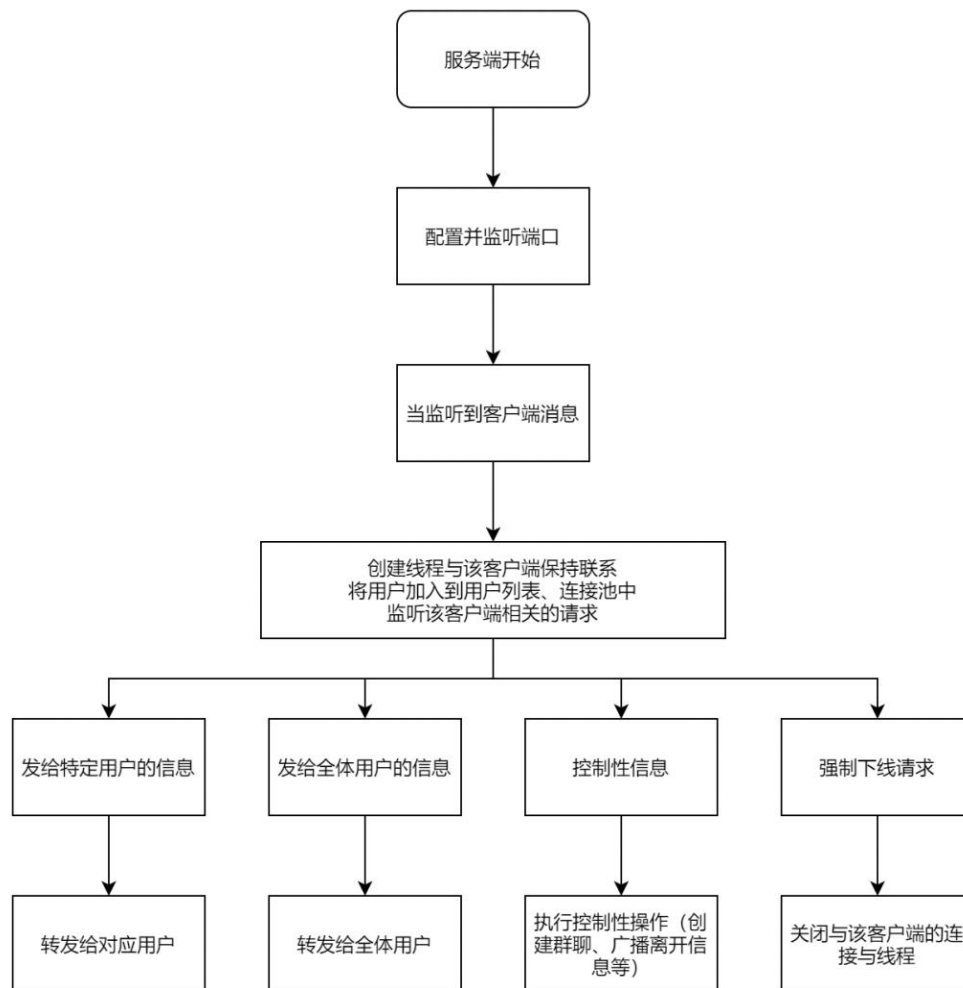
- 客户端的上线下线状态可以存储在服务器端的内存中，使用数据结构如列表或映射来管理在线状态。

- 当客户端上线或下线时，服务器更新相应客户端的在线状态，并广播给其他客户端，以便其他客户端能够实时刷新在线状态。

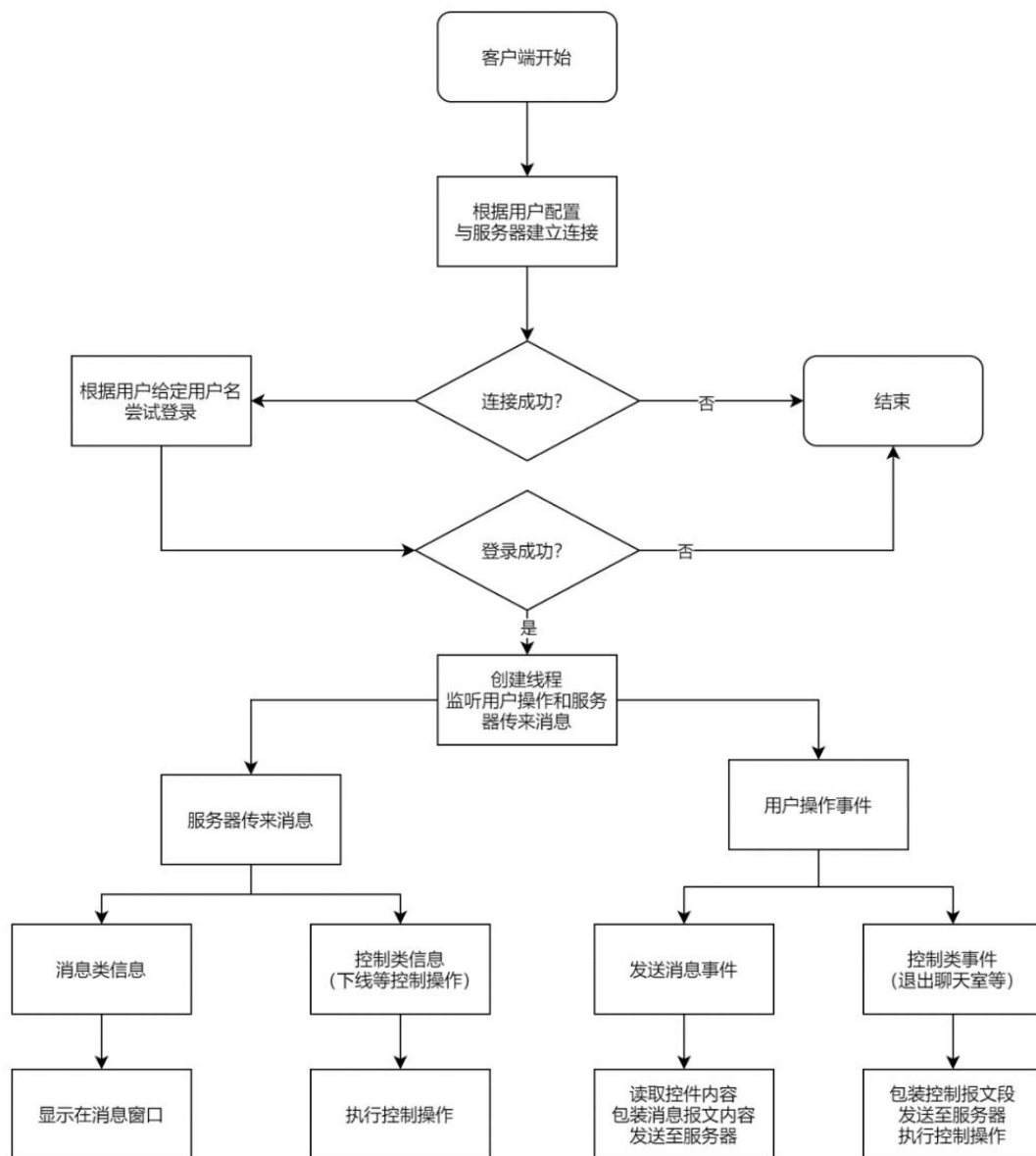
三、网络聊天程序系统设计

3.1 聊天室设计流程图

3.1.1 聊天室服务器端流程图



3.1.2 聊天室客户端流程图



3.2 关键问题设计

3.2.1 客户端创建主窗体并连接服务器

1. 连接服务器

如果服务器没有打开，则返回连接失败的错误信息

2. 创建登录窗口

3.2.2 客户端注册、登录

1. 注册窗口

注册成功后，向服务器发送一个请求，服务器将新用户信息正确存储后，向用户返回一个 OK 状态，然后系统将提示用户注册成功

2. 登录窗口

用户登录时，正确输入账号密码后，将向服务器发出一个 login 消息，服务器此时要判断用户是否已经登录，即判断用户 ID 是否已经在线用户列表中，如果已经登录，将向客户端发送提示消息，否则即为正确登录，打开聊天窗体

3.2.3 客户端私聊、公聊，收发文件，发送表情，发送窗口震动

1. 私聊、公聊

通过是否选中判断私发信息还是群发信息，先在本人界面显示发送的消息，再向服务器发送一个 chat 信息，然后给发送的对象（单个或多个）返回响应

2. 发送文件

和发送信息类似，利用 JFileChooser 和自己设计的 FileInfo 实现对文件信息的打包，并发送 toSendFile 给服务器

3. 发送表情

会记录表情的位置以及编号，会在和文本信息一起打包成消息给服务器发送过去显示的原理就是找到图片在文本中的位置，并将图片路径所在插入到窗口，完成显示

4. 发送窗口振动

和发送信息同理，向服务器发送一个名为 shake 的请求

5. 窗口振动实现函数

3.2.4 客户端历史记录

历史记录因为是针对用户的，所以只在客户端实现。实现原理即为在向消息面板更新消息时把消息同步保存在本地 txt 里即可

3.2.5 服务器创建主窗体

先初始化服务器套节字，再开启线程监听客户端的连接请求，然后启动服务器监控窗体

3.2.6 服务器私发消息、广播、踢人、禁言及解除禁言

1. 发送消息

服务器会以 **admin** 的名字来发送消息，直接给对应客户端发送 **chat** 响应

2. 广播

给所有在线用户发送 **bored** 的响应

3. 踢人

给对应客户端发送 **remove** 响应，客户端接收响应后，会实现 **logout**，和自己主动下线时一样的函数

4. 禁言及解除禁言

当服务器给客户端发送 **sil** 或者 **uns** 响应后，客户端接收 **sil** 会将聊天框的四个按钮变为灰色，反之则会变亮

四、网络聊天程序源代码清单

4.1 客户端创建主窗体并连接服务器

4.1.1 连接服务器

```
public static void connection() {
    String ip = "127.0.0.1";//获取服务器 ip 地址
    int port = 6666;//获取服务器为其分配到端口号
    try {
        DataBuffer.clientSocket = new Socket(ip, port);//根据 ip 地址和端口号
        新建 socket 套接字
        DataBuffer.oos = new ObjectOutputStream(DataBuffer.clientSocket.get
        OutputStream());
        DataBuffer.ois = new ObjectInputStream(DataBuffer.clientSocket.getI
        nputStream());

        } catch (Exception e) {
            JOptionPane.showMessageDialog(new JFrame(),
                "连接服务器失败, 请检查!", "服务器未连上
", JOptionPane.ERROR_MESSAGE);//否则连接失败
            System.exit(0);
        }
    }
}
```

如果服务器没有打开，则返回连接失败的错误信息

4.1.2 创建登录窗口

```
JFrame.setDefaultLookAndFeelDecorated(true);
JDialog.setDefaultLookAndFeelDecorated(true);
try {
    //UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFee
    lClassName());
    String lookAndFeel = UIManager.getSystemLookAndFeelClassName();
    UIManager.setLookAndFeel(lookAndFeel);
    //      UIManager.setLookAndFeel(new SubstanceBusinessLookAndFeel());

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
new LoginFrame(); //启动登录窗体
```

4.2 客户端注册、登录

4.2.1 注册窗口

```
private void registe(User user) throws IOException, ClassNotFoundException{
    Request request = new Request();
    request.setAction("userRegiste");
    request.setAttribute("user", user);
    //获取响应
    Response response = ClientUtil.sendTextRequest(request);
    ResponseStatus status = response.getStatus();
    switch(status){
        case OK:
            User user2 = (User)response.getData("user");
            JOptionPane.showMessageDialog(RegisterFrame.this,
                "您注册成功，您的账号为:"+ user2.getId()+"请牢记!" ,
                "注册成功",JOptionPane.INFORMATION_MESSAGE);
            this.setVisible(false);
            break;
        default:
            JOptionPane.showMessageDialog(RegisterFrame.this,
                "注册失败，请稍后再试！","服务器内部错误！
            ",JOptionPane.ERROR_MESSAGE);
    }
}
```

注册成功后，向服务器发送一个请求，服务器将新用户信息正确存储后，向用户返回一个 OK 状态，然后系统将提示用户注册成功

4.2.2 登录窗口

```
private void login() {
    if (idTxt.getText().length() == 0
        || pwdFld.getPassword().length == 0) {
        JOptionPane.showMessageDialog(LoginFrame.this,
            "账号和密码是必填的",
            "输入有误",JOptionPane.ERROR_MESSAGE);
        idTxt.requestFocusInWindow();
    }
}
```

```

        return;
    }

    if(!idTxt.getText().matches("\\d+")){
        JOptionPane.showMessageDialog(LoginFrame.this,
            "账号必须是数字",
            "输入有误",JOptionPane.ERROR_MESSAGE);
        idTxt.requestFocusInWindow();
        return;
    }

    Request req = new Request();
    //请求的动作为 userlogin
    req.setAction("userLogin");
    //request 请求域请求添加 id 和 password
    req.setAttribute("id", idTxt.getText());
    req.setAttribute("password", new String(pwdFld.getPassword()));

    //获取响应
    Response response = null;
    try {
        response = ClientUtil.sendTextRequest(req);//利用 ClientUtil 发生
request 并获取响应
    } catch (IOException e1) {
        e1.printStackTrace();
    }
    if(response.getStatus() == ResponseStatus.OK){
        //获取当前用户
        User user2 = (User)response.getData("user");
        if(user2!= null){ //登录成功
            //获取当前客户端的 user 信息
            DataBuffer.currentUser = user2;
            //获取当前在线用户列表
            DataBuffer.onlineUsers = (List<User>)response.getData("onlineUs
ers");

            LoginFrame.this.dispose();//关闭登录界面
            new ChatFrame(); //打开聊天窗体
        }else{ //登录失败
            String str = (String)response.getData("msg");
            JOptionPane.showMessageDialog(LoginFrame.this,
                str,
                "登录失败",JOptionPane.ERROR_MESSAGE);
        }
    }

```

```

    }else{
        JOptionPane.showMessageDialog(LoginFrame.this,
            "服务器内部错误，请稍后再试!!!", "登录失败",
            JOptionPane.ERROR_MESSAGE);
    }
}

```

用户登录时，正确输入账号密码后，将向服务器发出一个 login 消息，服务器此时要判断用户是否已经登录，即判断用户 ID 是否已经在在线用户列表中，如果已经登录，将向客户端发送提示消息，否则即为正确登录，打开聊天窗体

4.3 客户端私公聊，收发文件，发送表情，发送窗口震动

4.3.1 私聊、公聊

```

public void sendTxtMsg(){
    String content = this.sendArea.getText();//取文本消息
    String sentmessage=this.sendArea.getText()+'*'+buildPicInfo();
    System.out.println("content"+content);
    if ("".equals(content)) { //无内容
        JOptionPane.showMessageDialog(ChatFrame.this, "不能发送空消息!",
            "不能发送", JOptionPane.ERROR_MESSAGE);
    } else { //发送
        User selectedUser = (User)onlineList.getSelectedValue();
        //如果设置了 ToUser 表示私聊，否则群聊
        Message msg = new Message();
        if(rybqBtn.isSelected()){ //私聊
            if(null == selectedUser){
                JOptionPane.showMessageDialog(ChatFrame.this, "没有选择私聊
对象!",
                    "不能发送", JOptionPane.ERROR_MESSAGE);
                return;
            }else if (DataBuffer.currentUser.getId() == selectedUser.getId(
)){
                JOptionPane.showMessageDialog(ChatFrame.this, "不能给自己发
送消息!",
                    "不能发送", JOptionPane.ERROR_MESSAGE);
                return;
            }else{
                msg.setToUser(selectedUser);
            }
        }
    }
}

```

```

    }
    msg.setFromUser(DataBuffer.currentUser);
    msg.setSendTime(new Date());

    DateFormat df = new SimpleDateFormat("HH:mm:ss");
    StringBuffer sb = new StringBuffer();
    sb.append(" ").append(df.format(msg.getSendTime())).append(" ")
        .append(msg.getFromUser().getNickname())
        .append("(").append(msg.getFromUser().getId()).append(")");
    if(!this.rybqBtn.isSelected()){//群聊
        sb.append("对大家说");
    }else{
        sb.append("对
    ").append(msg.getToUser().getNickname()).append("(").append(msg.getToUser()
        .getId()).append(")");
    }
    SimpleAttributeSet attrib = new SimpleAttributeSet();
    try { // 插入文本

        msgListArea.getStyledDocument().insertString(msgListArea.getSty
ledDocument().getLength(), sb + "\n",
            attrib);
        msgListArea.setCaretPosition(msgListArea.getStyledDocument().ge
tLength()); // 设置滚动到最下边
    } catch (BadLocationException e) {
        e.printStackTrace();
    }
    int pos=msgListArea.getCaretPosition();
    sb.append("\n").append(sentmessage).append("\n");
    msg.setMessage(sb.toString());
    saveMessage(DataBuffer.currentUser, msg.getMessage());
    /*封装请求对象*/
    Request request = new Request();
    request.setAction("chat");
    request.setAttribute("msg", msg);
    try {
        ClientUtil.sendTextRequest2(request);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

//JTextArea 中按“Enter”时，清空内容并回到首行
InputMap inputMap = sendArea.getInputMap();
ActionMap actionMap = sendArea.getActionMap();

```

```

Object transferTextActionKey = "TRANSFER_TEXT";
inputMap.put(KeyStroke.getKeyStroke(KeyEvent.VK_ENTER,0),transferTextActionKey);
actionMap.put(transferTextActionKey,new AbstractAction() {
    private static final long serialVersionUID = 704184194583059022
9L;

    public void actionPerformed(ActionEvent e) {
        sendArea.setText("");
        sendArea.requestFocus();
    }
});
sendArea.setText("");

try { // 插入文本
    msgListArea.getStyledDocument().insertString(msgListArea.getStyledDocument().getLength(), content + "\n",
        attrib);
    msgListArea.setCaretPosition(msgListArea.getStyledDocument().getLength()); // 设置滚动到最下边
} catch (BadLocationException e) {
    e.printStackTrace();
}
for (int i = 0; i < myPicInfo.size(); i++) {
    PicInfo pic = myPicInfo.get(i);
    msgListArea.setCaretPosition(pos+pic.getPos()); /*设置插入位置*/
    String fileName;
    fileName = "qqdefaultface/" + pic.getVal() + ".gif";/*修改图片路径*/
    msgListArea.insertIcon(new ImageIcon(PicsJWindow.class.getResource(fileName))); /*插入图片*/
}
myPicInfo.clear();
//把光标定位到文本域的最后一行
msgListArea.setCaretPosition(msgListArea.getDocument().getLength())
;

//ClientUtil.appendTxt2MsgListArea(msg.getMessage());
}
}

```

通过是否选中判断私发信息还是群发信息，先在本人界面显示发送的消息，再向服务器发生一个 chat 信息，然后给发送的对象（单个或多个）返回响应

4.3.2 发送文件

```

if(selectedUser != null){

```



```

        if(DataBuffer.currentUser.getId() == selectedUser.getId()){
            JOptionPane.showMessageDialog(ChatFrame.this, "不能给自己发送文件!",
                "不能发送", JOptionPane.ERROR_MESSAGE);
        }else{
            JFileChooser jfc = new JFileChooser();
            if (jfc.showOpenDialog(ChatFrame.this) == JFileChooser.APPROVE_OPTI
ON) {
                File file = jfc.getSelectedFile();
                sendFile = new FileInfo();
                sendFile.setFromUser(DataBuffer.currentUser);
                sendFile.setToUser(selectedUser);
                try {
                    sendFile.setSrcName(file.getCanonicalPath());
                } catch (IOException e1) {
                    e1.printStackTrace();
                }
                sendFile.setSendTime(new Date());

                Request request = new Request();
                request.setAction("toSendFile");
                request.setAttribute("file", sendFile);
                try {
                    ClientUtil.sendTextRequest2(request);
                } catch (IOException e) {
                    e.printStackTrace();
                }
                saveMessage(sendFile.getFromUser(), "【文件消息】向 "
                    + selectedUser.getNickname() + "("
                    + selectedUser.getId() + ") 发送文件 ["
                    + file.getName() + "], 等待对方接收...\n");
                ClientUtil.appendTxt2MsgListArea("【文件消息】向 "
                    + selectedUser.getNickname() + "("
                    + selectedUser.getId() + ") 发送文件 ["
                    + file.getName() + "], 等待对方接收...\n");
            }
        }
    }else{
        JOptionPane.showMessageDialog(ChatFrame.this, "不能给所有在线用户发送文
件!",
            "不能发送", JOptionPane.ERROR_MESSAGE);
    }
}

```

和发送信息类似, 利用 JFileChooser 和自己设计的 FileInfo 实现对文件信息的

打包，并发送 toSendFile 给服务器

4.3.3 发送表情

```
try { // 插入文本
    msgListArea.getStyledDocument().insertString(msgListArea.getStyledDocument().getLength(), content + "\n",
        attrib);
    msgListArea.setCaretPosition(msgListArea.getStyledDocument().getLength()); // 设置滚动到最下边
} catch (BadLocationException e) {
    e.printStackTrace();
}
for (int i = 0; i < myPicInfo.size(); i++) {
    PicInfo pic = myPicInfo.get(i);
    msgListArea.setCaretPosition(pos+pic.getPos()); /*设置插入位置*/
    String fileName;
    fileName = "qqdefaultface/" + pic.getVal() + ".gif"; /*修改图片路径*/
    msgListArea.insertIcon(new ImageIcon(PicsJWindow.class.getResource(fileName))); /*插入图片*/
}
```

会记录表情的位置以及编号，会在和文本信息一起打包成消息给服务器发送过去显示的原理就是找到图片在文本中的位置，并将图片路径所在插入到窗口，完成显示

4.3.4 发送窗口振动

```
User selectedUser = (User)onlineList.getSelectedValue();
if(null != selectedUser){
    if(DataBuffer.currentUser.getId() == selectedUser.getId()){
        JOptionPane.showMessageDialog(ChatFrame.this, "不能给自己发送振动!",
            "不能发送", JOptionPane.ERROR_MESSAGE);
    }else{
        Message msg = new Message();
        msg.setFromUser(DataBuffer.currentUser);
        msg.setToUser(selectedUser);
        msg.setSendTime(new Date());

        DateFormat df = new SimpleDateFormat("HH:mm:ss");
        StringBuffer sb = new StringBuffer();
        sb.append(" ").append(msg.getFromUser().getNickname())
            .append("(").append(msg.getFromUser().getId()).append(" ")
```

```

        .append(df.format(msg.getSendTime()))
        .append("\n 给").append(msg.getToUser().getNickname())
        .append("(").append(msg.getToUser().getId()).append(" ")
        .append("发送了一个窗口抖动\n");
msg.setMessage(sb.toString());

Request request = new Request();
request.setAction("shake");
request.setAttribute("msg", msg);
try {
    ClientUtil.sendTextRequest2(request);
} catch (IOException e) {
    e.printStackTrace();
}
saveMessage(msg.getFromUser(), msg.getMessage());
ClientUtil.appendTxt2MsgListArea(msg.getMessage());
new JFrameShaker(ChatFrame.this).startShake();
}
}else{
    JOptionPane.showMessageDialog(ChatFrame.this, "不能群发送振动!",
        "不能发送", JOptionPane.ERROR_MESSAGE);
}

```

和发送信息同理，向服务器发送一个名为 shake 的请求

```

public void startShake() {
    oldLocation = frame.getLocation(); // 获取窗口的原始位置

    startTime = System.currentTimeMillis(); // 开始计时
    shakeTimer = new Timer((int) SHAKE_CYCLE / 5, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            long elapsed = System.currentTimeMillis() - startTime; // 计算振
动的用时
            //System.out.println(elapsed);
            // 利用时间计算出某一时刻晃动的幅度
            double waveOffset = (elapsed % SHAKE_CYCLE)*10;
            double angle = waveOffset * Math.PI;
            double angleY = waveOffset * Math.PI;
            int shakeX = (int) ((Math.sin(angle) * SHAKE_DISTANCE) + oldLoc
ation.x);
            int shakeY = (int) ((Math.sin(angleY) * SHAKE_DISTANCE) + oldLo
cation.y);
            frame.setLocation(shakeX, shakeY);
            if (elapsed >= SHAKE_DURATION) { // 振动时长到了就停止

```

```

        stopShake();
    }
}
});
shakeTimer.start(); // 启动定时任务
}

```

窗口振动实现函数

4.4 客户端历史记录

```

public static void saveMessage(User user,String txt){
    if (user != null) System.out.println(user);
    else System.out.println("null");
    System.out.println(txt);
    if(user != null) {
        try {
            File f = new File(user.getNickname()+"的历史聊天记录.txt");
            FileWriter fw = new FileWriter(f, true);
            BufferedWriter bw = new BufferedWriter(fw);
            bw.write(txt);
            bw.newLine();
            bw.flush();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

历史记录因为是针对用户的，所以只在客户端实现。实现原理即为在向消息面板更新消息时把消息同步保存在本地 txt 里即可

4.5 服务器创建主窗体

```

int port = 6666;
//初始化服务器套节字
try {
    DataBuffer.serverSocket = new ServerSocket(port);
} catch (IOException e) {
    e.printStackTrace();
}

```

```

new Thread(new Runnable() { //启动新线程进行客户端连接监听
    public void run() {
        try {
            while (true) {
                // 监听客户端的连接
                Socket socket = DataBuffer.serverSocket.accept();
                System.out.println("客户来了: "
                    + socket.getInetAddress().getHostAddress()
                    + ":" + socket.getPort());

                //针对每个客户端启动一个线程，在线程中调用 RequestProcessor
                //来处理每个客户端的请求
                new Thread(new RequestProcessor(socket)).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}).start();

//设置外观感觉
JFrame.setDefaultLookAndFeelDecorated(true);
JDialog.setDefaultLookAndFeelDecorated(true);
try {
    UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
} catch (Exception e) {
    e.printStackTrace();
}
//启动服务器监控窗体
new ServerInfoFrame();
}

```

先初始化服务器套节字，再开启线程监听客户端的连接请求，然后启动服务器监控窗体

4.6 服务器私发消息、广播、踢人、禁言及解除禁言

4.6.1 发送消息

```

public static void chat_sys(String str, User user_) throws IOException{
    User user = new User(1, "admin");
    Message msg = new Message();
}

```

```

msg.setFromUser(user);
msg.setSendTime(new Date());
msg.setToUser(user_);

DateFormat df = new SimpleDateFormat("HH:mm:ss");
StringBuffer sb = new StringBuffer();
sb.append(" ").append(df.format(msg.getSendTime())).append(" ");
sb.append("系统通知\n "+str+"\n");
msg.setMessage(sb.toString());

Response response = new Response();
response.setStatus(ResponseStatus.OK);
response.setType(ResponseType.CHAT);
response.setData("txtMsg", msg);

OnlineClientIOCache io = DataBuffer.onlineUserIOCacheMap.get(msg.getToUser().getId());
sendResponse_sys(io, response);
}

```

服务器会以 admin 的名字来发送消息，直接给对应客户端发送 chat 响应

4.6.2 广播

```

public static void board(String str) throws IOException {
    User user = new User(1,"admin");
    Message msg = new Message();
    msg.setFromUser(user);
    msg.setSendTime(new Date());

    DateFormat df = new SimpleDateFormat("HH:mm:ss");
    StringBuffer sb = new StringBuffer();
    sb.append(" ").append(df.format(msg.getSendTime())).append(" ");
    sb.append("系统通知\n "+str+"\n");
    msg.setMessage(sb.toString());

    Response response = new Response();
    response.setStatus(ResponseStatus.OK);
    response.setType(ResponseType.BOARD);
    response.setData("txtMsg", msg);

    for (Long id : DataBuffer.onlineUserIOCacheMap.keySet()) {
        sendResponse_sys(DataBuffer.onlineUserIOCacheMap.get(id), response)
    }
}

```

```
}
```

给所有在线用户发送 bored 的响应

4.6.3 踢人

```
public static void remove(User user_) throws IOException{
    User user = new User(1,"admin");
    Message msg = new Message();
    msg.setFromUser(user);
    msg.setSendTime(new Date());
    msg.setToUser(user_);

    StringBuffer sb = new StringBuffer();
    DateFormat df = new SimpleDateFormat("HH:mm:ss");
    sb.append(" ").append(df.format(msg.getSendTime())).append(" ");
    sb.append("系统通知您\n "+"您被强制下线"+"\\n");
    msg.setMessage(sb.toString());

    Response response = new Response();
    response.setStatus(ResponseStatus.OK);
    response.setType(ResponseType.REMOVE);
    response.setData("txtMsg", msg);

    OnlineClientIOCache io = DataBuffer.onlineUserIOCacheMap.get(msg.getToU
ser().getId());
    sendResponse_sys(io, response);
}
```

给对应客户端发送 remove 响应，客户端接收响应后，会实现 logout，和自己主动下线时一样的函数

4.6.4 禁言及解除禁言

```
public static void silence(User user_) throws IOException{
    User user = new User(1,"admin");
    Message msg = new Message();
    msg.setFromUser(user);
    msg.setSendTime(new Date());
    msg.setToUser(user_);

    DateFormat df = new SimpleDateFormat("HH:mm:ss");
    StringBuffer sb = new StringBuffer();
```

```

sb.append(" ").append(df.format(msg.getSendTime())).append(" ");
sb.append("系统通知您\n  "+"您已被禁言"+"\\n");
msg.setMessage(sb.toString());

Response response = new Response();
response.setStatus(ResponseStatus.OK);
response.setType(ResponseType.SILENCE);
response.setData("txtMsg", msg);

OnlineClientIOCache io = DataBuffer.onlineUserIOCacheMap.get(msg.getToU
ser().getId());
sendResponse_sys(io, response);
}
/*解除禁言用户*/
public static void unsilence(User user_) throws IOException{
    User user = new User(1,"admin");
    Message msg = new Message();
    msg.setFromUser(user);
    msg.setSendTime(new Date());
    msg.setToUser(user_);

    StringBuffer sb = new StringBuffer();
    DateFormat df = new SimpleDateFormat("HH:mm:ss");
    sb.append(" ").append(df.format(msg.getSendTime())).append(" ");
    sb.append("系统通知您\n  "+"您被解除禁言"+"\\n");
    msg.setMessage(sb.toString());

    Response response = new Response();
    response.setStatus(ResponseStatus.OK);
    response.setType(ResponseType.UNSILENCE);
    response.setData("txtMsg", msg);

    OnlineClientIOCache io = DataBuffer.onlineUserIOCacheMap.get(msg.getToU
ser().getId());
    sendResponse_sys(io, response);
}

```

当服务器给客户端发送 sil 或者 uns 响应后,客户端接收 sil 会将聊天框的四个按钮变为灰色,反之则会变亮

五、网络聊天程序运行结果与测试分析

5.1 注册登录界面

注册界面：



A screenshot of a Windows-style window titled "注册新账号" (Register New Account). The window has a standard title bar with minimize, maximize, and close buttons. The main content area contains three input fields: "昵称:" (Nickname), "密码:*" (Password), and "确认密码:*" (Confirm Password). To the right of the password fields is a "性别:" (Gender) section with two radio buttons labeled "男" (Male) and "女" (Female). At the bottom of the window are three buttons: "确认" (Confirm), "重填" (Re-enter), and "取消" (Cancel).

登录界面：



A screenshot of a Windows-style window titled "登录" (Login). The window has a standard title bar with minimize, maximize, and close buttons. The main content area has a header "输入登录信息" (Enter login information). Below this header are two input fields: "账号:" (Account) and "密码:" (Password). At the bottom of the window are two buttons: "注册" (Register) and "登录" (Login).

5.2 客户端私聊、公聊界面

公聊界面：



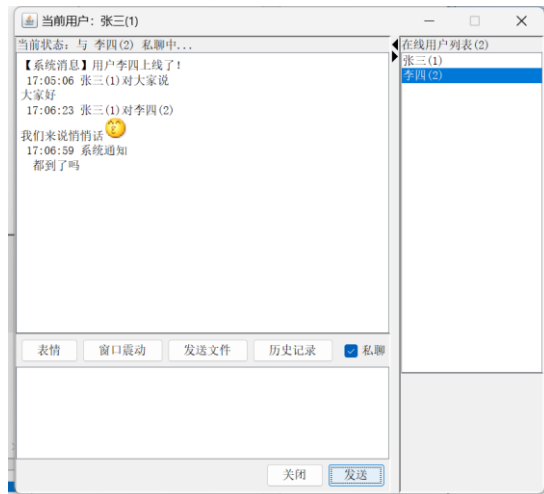
私聊界面：

张三与用户李四私聊，李四的界面：

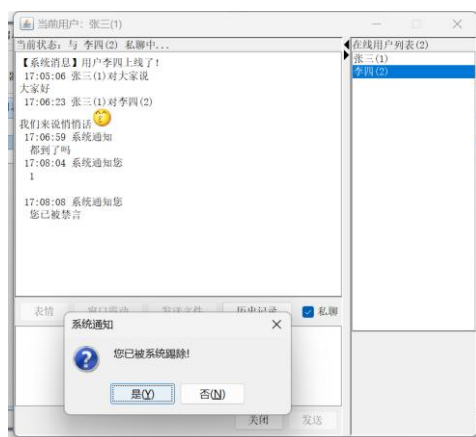


5.3 服务器发送系统消息、踢人、禁言界面

服务器广播系统消息：



服务器踢人：



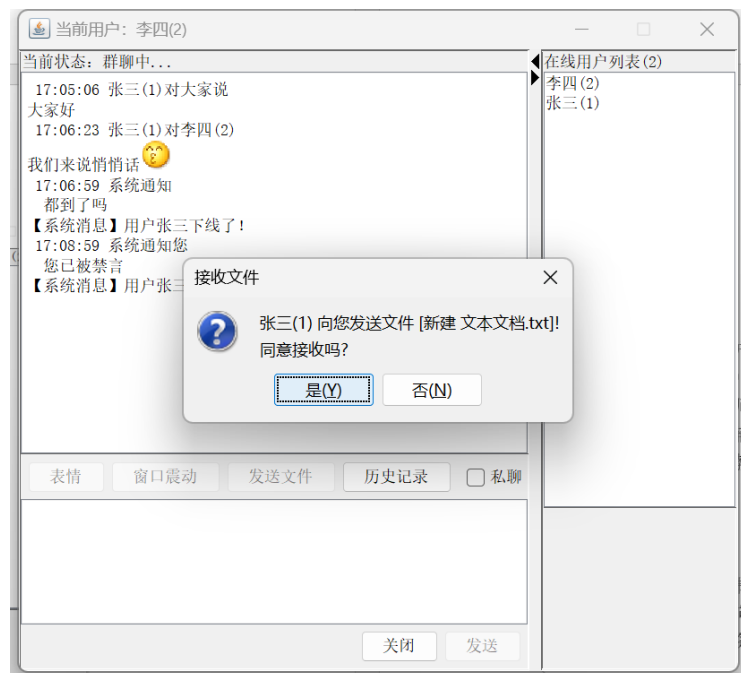
禁言：

系统会提示已禁言/解除禁言



5.4 发送文件

张三向用户李四发送文件，李四可以选择是否同意接受，接受后可以选择文件存放位置：

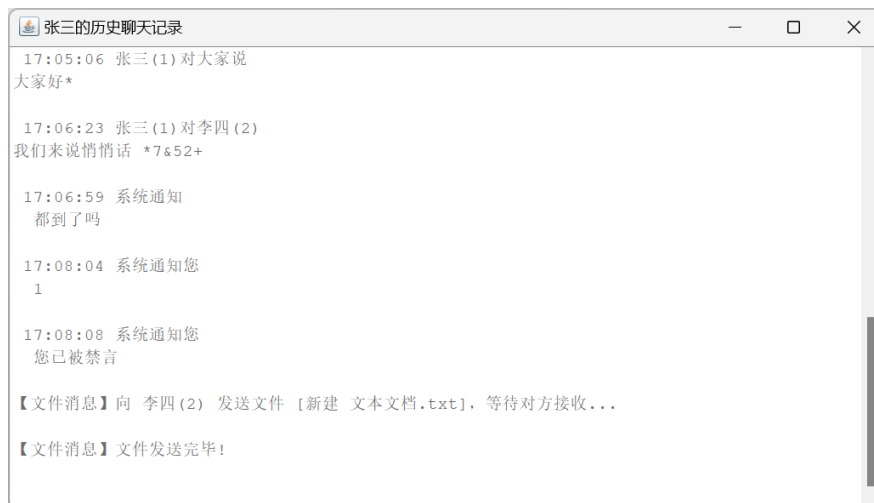


选好文件存放位置并点击确认后，提示用户保存成功：



5.5 客户端历史记录

点击历史记录, 就会弹出历史记录窗口:



六、爬取并分析网页敏感词程序业务分析

6.1 实验目的

用 Java URL 编程实现爬取并分析网页敏感词，其主要功能有：

1. 编写界面，输入一个网址，能够爬取该网址上所有的 HTML 源代码。
2. 对网址中的文本进行提取。
3. 建立敏感词库，用文本文件保存。
4. 将该网址所对应的文本中的敏感词提取并高亮显示。
5. 编写文本文件，可以存入多个网址；程序可爬取这些网址中的文本内容，将敏感词记录存入另一个文件，格式自定。
6. 编写一个主界面，整合上述功能

6.2 需求设计

1. 用户界面：
提供一个界面，用户可以输入一个网址。
有一个按钮，用于触发爬取网址功能。
2. 爬取网页功能：
根据用户输入的网址，爬取该网址上的所有 HTML 源代码。
3. 文本提取功能：
对爬取的网页内容进行文本提取，提取出网页中的文本信息。
4. 敏感词库：
建立一个敏感词库，将敏感词保存在一个文本文件中。
5. 敏感词提取和高亮显示：
在提取出的网页文本中，识别并高亮显示敏感词。
6. 文本文件管理：
允许用户创建文本文件，将多个网址存入其中。
程序可以读取文本文件中的网址列表，爬取这些网址中的文本内容。

将提取到的敏感词记录存入另一个文件，格式由程序定义。

7. 主界面整合：

编写一个主界面，整合上述功能。

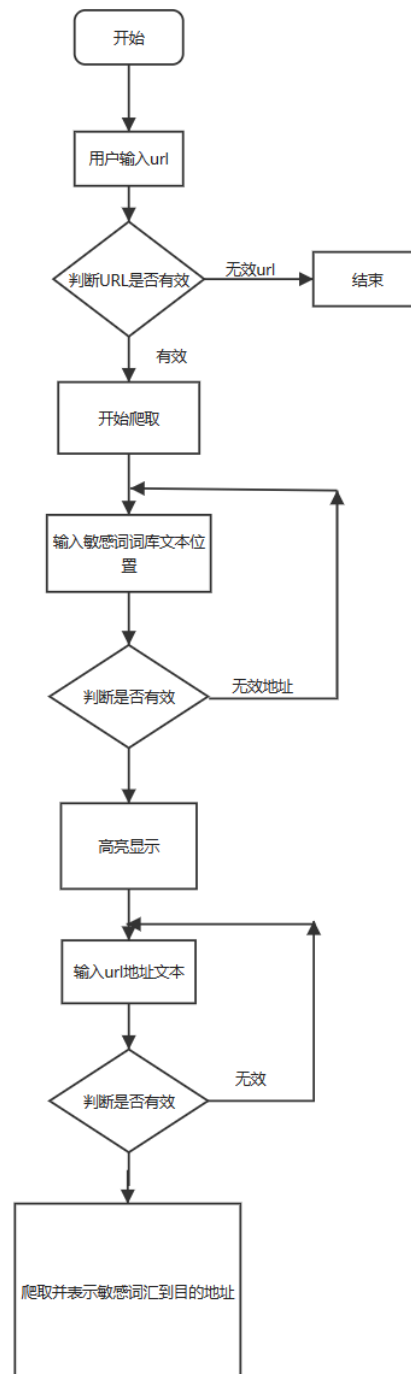
在主界面上展示用户输入网址的输入框、爬取按钮、敏感词高亮显示区域以及其他必要的功能模块。

6.3 实现思路：

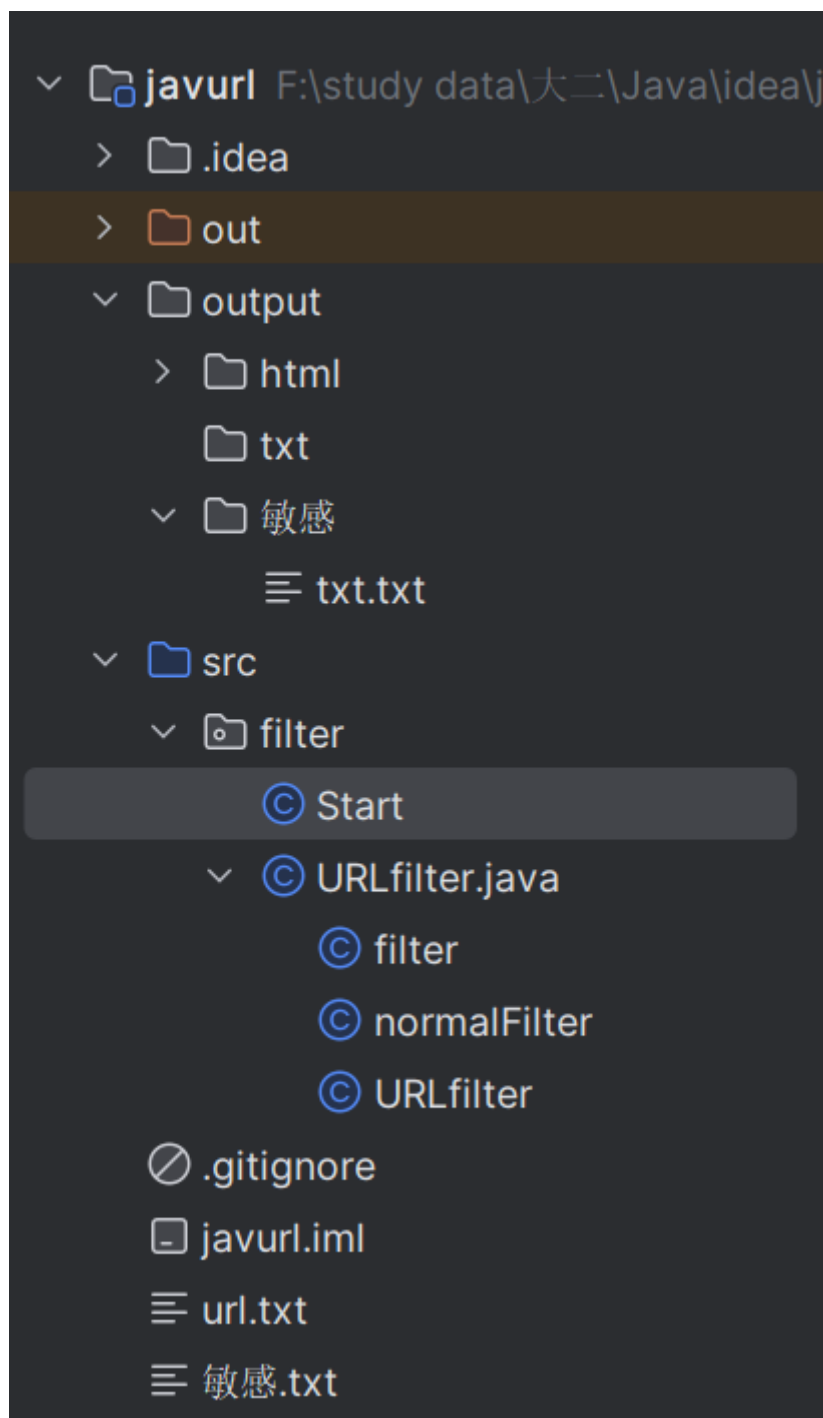
1. 使用 Java 编程语言编写一个图形用户界面(GUI)程序，可以接收用户输入的网址。
2. 使用 Java 的 URL 类库实现网页爬取功能，获取网页的 HTML 源代码。
3. 利用 HTML 解析库（如 Jsoup）对获取的 HTML 源代码进行解析，提取其中的文本内容。
4. 实现敏感词库的建立和管理功能，将敏感词保存在一个文本文件中。
5. 在提取到的文本内容中识别敏感词，并进行高亮显示。
6. 实现文本文件的读写功能，能够将多个网址存入文件中，并将敏感词记录存入另一个文件。
7. 设计一个主界面，整合上述功能模块，让用户可以方便地输入网址、进行爬取和敏感词分析。

七、爬取并分析网页敏感词程序系统设计

7.1 爬取并分析网页敏感词流程图



7.2 爬取并分析网页敏感词代码文件



运行后，文件读取 `url.txt` 和 `敏感.txt`，将文件输出到 `/敏感/txt.txt`；

八、爬取并分析网页敏感词源代码清单

8.1 Start.java

用来实现最基础的 UI

```
package filter;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;

public class Start extends JFrame implements ActionListener{
    /*      GUI      */
    private JTextArea Text = new JTextArea();//聊天页面
    private JTextField InputText = new JTextField();//输入页面
    private JButton send = new JButton();//发送按钮
    public Start(){

        int width = 1000;
        int height = 800;
        /*
        可视化界面
        */
        this.setTitle("登录");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(width,height);//页面
        this.setVisible(true);
        this.setLayout(null);

        /*
        文本页面
        */
        Text.setBackground(Color.white);
        Text.setBounds(0,0, (int) (width),(int) (height*0.8));
        this.add(Text);

        /*
        输入页面
        */
        InputText.setBackground(Color.lightGray);
```

```

        InputText.setBounds(0,(int) (height*0.8),(int) (width*0.8),(int)(height*0.2));
        this.add(InputText);
        /*
        按钮
        */
        send.setBackground(Color.PINK);
        send.setText("Send");
        send.setBounds((int) (width*0.8),(int)(height*0.8),(int)(width*0.2)
,(int)(height*0.2));
        this.add(send);

        send.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
            }
        });

    }

    @Override
    public void actionPerformed(ActionEvent e) {

    }
}

```

8.2 URLfilter. java

实现最主要的功能

8.2.1 Filter. java:

```

package filter;

import java.io.InputStream;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Rectangle;

```

```

import java.awt.Shape;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.Console;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.lang.reflect.Field;
import java.net.URL;
import java.text.NumberFormat;
import java.util.HashMap;
import java.util.Iterator;
import java.util.TimerTask;
import java.util.Vector;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.naming.spi.DirStateFactory.Result;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.text.JTextComponent;
import javax.swing.text.DefaultHighlighter.DefaultHighlightPainter;
import javax.swing.text.Highlighter;
import javax.swing.text.Highlighter.Highlight;
import javax.swing.text.Highlighter.HighlightPainter;

public class URLfilter extends JFrame {
    public static void main(String[] args) {
        new normalFilter();
    }
}

```

```
}  
}
```

```
class normalFilter extends JFrame {  
    JPanel mainPanel = new JPanel();  
    JPanel buttonPanel = new JPanel();  
    JButton jButton1 = new JButton("导入敏感词库");  
    JButton jButton2 = new JButton("高亮显示敏感词");  
    JButton jButton3 = new JButton("爬取目标 URL");  
    JButton jButton4 = new JButton("导入目标 URL 库");  
    JButton jButton5 = new JButton("开始批量爬取");  
    JLabel jLabel = new JLabel("请导入敏感词库");  
    JLabel urLabel = new JLabel("请导入目标 URL 库");  
    JTextArea jTextArea = new JTextArea();  
    JScrollPane jScrollPane = new JScrollPane(jTextArea);  
    JProgressBar jProgressBar = new JProgressBar();  
    normalFilter self = this;  
    String fileName = null;  
    String source = null;  
    String outputPath = null;  
    Vector<String> urlvector = null;  
    Vector<String> wordvector = null;  
    AtomicInteger count = new AtomicInteger(0);  
  
    public normalFilter() {  
        this.setTitle("URL Filter");  
        this.setSize(800, 600);  
        this.setLocation(200, 200);  
        this.setVisible(true);  
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);  
        this.add(mainPanel);  
        mainPanel.setLayout(new BorderLayout());  
        mainPanel.add(jScrollPane, BorderLayout.CENTER);  
        mainPanel.add(buttonPanel, BorderLayout.SOUTH);  
        buttonPanel.setLayout(new GridLayout(4, 2));  
        buttonPanel.add(jButton3);  
        buttonPanel.add(jButton2);  
        buttonPanel.add(jButton1);  
        buttonPanel.add(jLabel);  
        buttonPanel.add(jButton4);  
        buttonPanel.add(urLabel);  
        buttonPanel.add(jButton5);  
        buttonPanel.add(jProgressBar);  
        jTextArea.setLineWrap(true);  
    }  
}
```

```

jTextArea.setWrapStyleWord(true);
jTextArea.setEditable(false);
jButton2.setEnabled(false);
jButton4.setEnabled(false);
jButton5.setEnabled(false);

```

```

jButton1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Vector<JButton> bVector = new Vector<>();
        bVector.add(jButton2);
        bVector.add(jButton4);
        wordvector = filter.importLib(bVector, jLabel1, self);
        jButton5.setEnabled(false);
    }
});

```

```

jButton2.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        filter.highLight(jTextArea, wordvector);
    }
});

```

```

jButton3.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        String source = JOptionPane.showInputDialog(self, "请输入目
标 URL");

        // 获取 url
        jButton3.setEnabled(false);
        new Thread(new Runnable() {

            @Override
            public void run() {
                try {
                    try {
                        // 创建 URL 对象，将网络资源路径传递到该对象进行
                        URL url2 = new URL(source);
                        // 通过 url 对象打开并且激活网络流来获取网页资源

```

绑定

```

        InputStream openStream = url2.openStream();
        InputStreamReader read = new InputStreamReader(
der(openStream, "UTF-8");

        BufferedReader br = new BufferedReader(read

);

        // 创建爬虫后要写入的文件
        FileOutputStream fos = new FileOutputStream
(new File("F:\\study data\\大二\\Java\\" +
            "idea\\javurl\\output\\html\\url.tx
t"));

        OutputStreamWriter osw = new OutputStreamWr
iter(fos);

        BufferedWriter bw = new BufferedWriter(osw)

;

        // 定义空变量来储存爬下来的内容
        String str = "";
        while ((str = br.readLine()) != null) {
            bw.write(str);
            bw.newLine();
            bw.flush();
        }
        // 关闭资源
        bw.close();
        osw.close();
        fos.close();
        br.close();
        read.close();
        openStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }

    jProgressBar.setIndeterminate(true);
    jProgressBar.setStringPainted(false);
    jTextArea.setText("");
    jTextArea.append(filter.deleteHTMLTag(filter.ge
tSource(source)));

    JOptionPane.showMessageDialog(self, source + "
爬取完成", "网页爬取完成", JOptionPane.PLAIN_MESSAGE);
} catch (Exception e2) {

```

```

        JOptionPane.showMessageDialog(self, "目标 URL 异常", "无法提取网页信息", JOptionPane.ERROR_MESSAGE);
        jButton2.setEnabled(false);
    }finally {
        jProgressBar.setIndeterminate(false);
        jProgressBar.setMaximum(1);
        jProgressBar.setValue(0);
        jProgressBar.setBorderPainted(false);
        jButton3.setEnabled(true);
    }
    }
    }).start();
}
});

jButton4.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Vector<JButton> bVector = new Vector<>();
        bVector.add(jButton5);
        urlvector = filter.importLib(bVector, urLabel, self);
    }
});

jButton5.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        jButton5.setEnabled(false);
        JFileChooser jFileChooser = new JFileChooser();
        jFileChooser.setCurrentDirectory(new File("."));
        jFileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_
ONLY);

        jFileChooser.showDialog(new Label(), "选择输出文件夹");
        outputPath = jFileChooser.getSelectedFile().getAbsolutePath
());

        if (outputPath == null) {
            return;
        }
        new Thread(new Runnable() {

            @Override
            public void run() {
                count.set(0);
                jProgressBar.setIndeterminate(false);

```



```

jProgressBar.setMinimum(0);
jProgressBar.setMaximum(urlvector.size());
jProgressBar.setValue(0);
jProgressBar.setStringPainted(true);
HashMap<Integer, Vector<String>> map = new HashMap<
>());

int pronum = 3;
for (int i = 0; i < pronum; i++) {
    map.put(i, new Vector<>());
}
for (int i = 0; i < urlvector.size(); i++) {
    int j = i % pronum;
    map.get(j).add(urlvector.elementAt(i));
}
for (int i = 0; i < pronum; i++) {
    if (map.get(i).size() != 0) {
        new scan(map.get(i), self);
    }
}
int c = 0;
AtomicInteger c1 = new AtomicInteger(0);
new Thread(new Runnable() {
    @Override
    public void run() {
        int c2;
        do {
            c2 = c1.get();
            Dimension dimension = jProgressBar.getSi
ize();

            Rectangle rectangle = new Rectangle(0,
0, dimension.width, dimension.height);
            jProgressBar.setValue(c2);
            jProgressBar.paintImmediately(rectangle
);

            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } while (c2 != urlvector.size());
    }
}).start();
while (true) {

```

```

        c = count.get();
        c1.set(c);
        if (c == urlvector.size()) {
            JOptionPane.showMessageDialog(self, "所有
URL 过滤完成", "过滤完成", JOptionPane.PLAIN_MESSAGE);
            jButton5.setEnabled(true);
            break;
        }
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {

        }
    }
}

}).start();
}

class scan implements Runnable {
    Vector<String> vector = null;
    normalFilter nFilter = null;

    public scan(Vector<String> v, normalFilter nf) {
        vector = v;
        nFilter = nf;
        new Thread(this).start();
    }

    @Override
    public void run() {
        for (String string : vector) {
            BufferedWriter bWriter = null;
            try {
                Pattern pattern = Pattern.compile("\\w+");
                Matcher matcher = pattern.matcher(string);
                String filename = "";
                while (matcher.find()) {
                    filename += matcher.group();
                }
                File file = new File(outputPath + "/" + filename
e + ".txt");

                if (!file.exists()) {
                    file.createNewFile();
                }
            }
        }
    }
}

```



```

    public static String deleteHTMLTag(String source) {
        String regEx_script = "<script[^>]*?>[\\s\\S]*?<\\/script>"; // 定义 script 的正则表达式
        String regEx_style = "<style[^>]*?>[\\s\\S]*?<\\/style>"; // 定义 style 的正则表达式
        String regEx_html = "<[^>]+>"; // 定义 HTML 标签的正则表达式
        String regEx_zhuanyi = "&[\\s\\S]*?"; // 定义转义字符&copy;等的正则表达式
        String regEx_space = "\\s{2,}|\\t";

        Pattern p_script = Pattern.compile(regEx_script, Pattern.CASE_INSENSITIVE);
        Matcher m_script = p_script.matcher(source);
        source = m_script.replaceAll(""); // 过滤 script 标签

        Pattern p_style = Pattern.compile(regEx_style, Pattern.CASE_INSENSITIVE);
        Matcher m_style = p_style.matcher(source);
        source = m_style.replaceAll(""); // 过滤 style 标签

        Pattern p_html = Pattern.compile(regEx_html, Pattern.CASE_INSENSITIVE);
        Matcher m_html = p_html.matcher(source);
        source = m_html.replaceAll(""); // 过滤 html 标签

        Pattern p_zhuanyi = Pattern.compile(regEx_zhuanyi, Pattern.CASE_INSENSITIVE);
        Matcher m_zhuanyi = p_zhuanyi.matcher(source);
        source = m_zhuanyi.replaceAll(""); // 过滤转义字符

        Pattern p_space = Pattern.compile(regEx_space, Pattern.CASE_INSENSITIVE);
        Matcher m_space = p_space.matcher(source);
        source = m_space.replaceAll("\\n");

        return source.trim(); // 返回文本字符串
    }

    public static void highLight(JTextArea jTextArea, Vector<String> vector) {
        String text = jTextArea.getText();
        int pos = 0;
        Highlighter highlighter = jTextArea.getHighlighter();

```

```

        DefaultHighlightPainter painter = new DefaultHighlightPainter(Color
.green);////////
        for (String string : vector) {
            while ((pos = text.indexOf(string, pos)) >= 0) {
                try {
                    highlighter.addHighlight(pos, pos + string.length(), pa
inter);

                    pos += string.length();
                } catch (Exception e) {

                }
            }
        }
    }
}

```

```

    public static Vector<String> readWords(String fileName) throws Exceptio
n {
        Vector<String> vector = new Vector<>();
        BufferedReader bReader = new BufferedReader(
            new InputStreamReader(new FileInputStream(new File(fileName
)), "GBK"));
        String line;
        while ((line = bReader.readLine()) != null) {
            vector.add(line);
        }
        return vector;
    }
}

```

```

    public static Vector<String> importLib(Vector<JButton> bVector, JLabel
jLabel, normalFilter self) {
        Vector<String> vector = new Vector<>();
        JFileChooser jFileChooser = new JFileChooser();
        jFileChooser.setCurrentDirectory(new File("."));
        jFileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
        jFileChooser.setAcceptAllFileFilterUsed(false);
        jFileChooser.addChoosableFileFilter(new FileFilter() {

            @Override
            public String getDescription() {
                // TODO Auto-generated method stub
                return "文本文件(*.txt)";
            }

            @Override

```

```

        public boolean accept(File f) {
            // TODO Auto-generated method stub
            return f.getName().endsWith("txt") || f.isDirectory();
        }
    });
    jFileChooser.showDialog(new Label(), "选择");
    try {
        String fileName = jFileChooser.getSelectedFile().getAbsolutePath();
        h();

        vector = filter.readWords(fileName);
        jLabel.setText("已导入目标库: " + fileName);
        for (JButton jButton : bVector) {
            jButton.setEnabled(true);
        }
    } catch (Exception e2) {
        new JOptionPane().showMessageDialog(self, "目标库文件错误", "ERROR", JOptionPane.ERROR_MESSAGE);
        jLabel.setText("请导入库文件");
        for (JButton jButton : bVector) {
            jButton.setEnabled(false);
        }
    }
    return vector;
}
}

```

8.2.2 URLfilter.java

```

package filter;

import java.io.InputStream;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.BufferedReader;

```

```

import java.io.BufferedWriter;
import java.io.Console;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.lang.reflect.Field;
import java.net.URL;
import java.text.NumberFormat;
import java.util.HashMap;
import java.util.Iterator;
import java.util.TimerTask;
import java.util.Vector;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.naming.spi.DirStateFactory.Result;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.text.JTextComponent;
import javax.swing.text.DefaultHighlighter.DefaultHighlightPainter;
import javax.swing.text.Highlighter;
import javax.swing.text.Highlighter.Highlight;
import javax.swing.text.Highlighter.HighlightPainter;

public class URLfilter extends JFrame {
    public static void main(String[] args) {
        new normalFilter();
    }
}

class normalFilter extends JFrame {

```

```

JPanel mainPanel = new JPanel();
JPanel buttonPanel = new JPanel();
JButton jButton1 = new JButton("导入敏感词库");
JButton jButton2 = new JButton("高亮显示敏感词");
JButton jButton3 = new JButton("爬取目标 URL");
JButton jButton4 = new JButton("导入目标 URL 库");
JButton jButton5 = new JButton("开始批量爬取");
JLabel jLabel = new JLabel("请导入敏感词库");
JLabel urLabel = new JLabel("请导入目标 URL 库");
JTextArea jTextArea = new JTextArea();
JScrollPane jScrollPane = new JScrollPane(jTextArea);
JProgressBar jProgressBar = new JProgressBar();
normalFilter self = this;
String fileName = null;
String source = null;
String outputPath = null;
Vector<String> urlvector = null;
Vector<String> wordvector = null;
AtomicInteger count = new AtomicInteger(0);

public normalFilter() {
    this.setTitle("URL Filter");
    this.setSize(800, 600);
    this.setLocation(200, 200);
    this.setVisible(true);
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.add(mainPanel);
    mainPanel.setLayout(new BorderLayout());
    mainPanel.add(jScrollPane, BorderLayout.CENTER);
    mainPanel.add(buttonPanel, BorderLayout.SOUTH);
    buttonPanel.setLayout(new GridLayout(4, 2));
    buttonPanel.add(jButton3);
    buttonPanel.add(jButton2);
    buttonPanel.add(jButton1);
    buttonPanel.add(jLabel);
    buttonPanel.add(jButton4);
    buttonPanel.add(urLabel);
    buttonPanel.add(jButton5);
    buttonPanel.add(jProgressBar);
    jTextArea.setLineWrap(true);
    jTextArea.setWrapStyleWord(true);
    jTextArea.setEditable(false);
    jButton2.setEnabled(false);
    jButton4.setEnabled(false);

```



```

jButton5.setEnabled(false);

jButton1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Vector<JButton> bVector = new Vector<>();
        bVector.add(jButton2);
        bVector.add(jButton4);
        wordvector = filter.importLib(bVector, jLabel1, self);
        jButton5.setEnabled(false);
    }
});

jButton2.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        filter.highLight(jTextArea, wordvector);
    }
});

jButton3.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        String source = JOptionPane.showInputDialog(self, "请输入目
标 URL");

        // 获取 url
        jButton3.setEnabled(false);
        new Thread(new Runnable() {

            @Override
            public void run() {
                try {
                    try {
                        // 创建 URL 对象，将网络资源路径传递到该对象进行
                        绑定

                        URL url2 = new URL(source);
                        // 通过 url1 对象打开并且激活网络流来获取网页资源
                        InputStream openStream = url2.openStream();
                        InputStreamReader read = new InputStreamRea
der(openStream, "UTF-8");

```

```

        BufferedReader br = new BufferedReader(read
);

        // 创建爬虫后要写入的文件
        FileOutputStream fos = new FileOutputStream
(new File("F:\\study data\\大二\\Java\\" +
        "idea\\javurl\\output\\html\\url.tx
t"));

        OutputStreamWriter osw = new OutputStreamWr
iter(fos);

        BufferedWriter bw = new BufferedWriter(osw)
;

        // 定义空变量来储存爬下来的内容
        String str = "";
        while ((str = br.readLine()) != null) {
            bw.write(str);
            bw.newLine();
            bw.flush();
        }
        // 关闭资源
        bw.close();
        osw.close();
        fos.close();
        br.close();
        read.close();
        openStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }

    jProgressBar.setIndeterminate(true);
    jProgressBar.setStringPainted(false);
    jTextArea.setText("");
    jTextArea.append(filter.deleteHTMLTag(filter.ge
tSource(source)));

    JOptionPane.showMessageDialog(self, source + "
爬取完成", "网页爬取完成", JOptionPane.PLAIN_MESSAGE);
} catch (Exception e2) {
    JOptionPane.showMessageDialog(self, "目标 URL 异
常", "无法提取网页信息", JOptionPane.ERROR_MESSAGE);
    jButton2.setEnabled(false);
}finally {

```

```

        progressBar.setIndeterminate(false);
        progressBar.setMaximum(1);
        progressBar.setValue(0);
        progressBar.setBorderPainted(false);
        jButton3.setEnabled(true);
    }
}
}).start();
}
});

jButton4.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Vector<JButton> bVector = new Vector<>();
        bVector.add(jButton5);
        urlvector = filter.importLib(bVector, urLabel, self);
    }
});

jButton5.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        jButton5.setEnabled(false);
        JFileChooser jFileChooser = new JFileChooser();
        jFileChooser.setCurrentDirectory(new File("."));
        jFileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_
ONLY);

        jFileChooser.showDialog(new Label(), "选择输出文件夹");
        outputPath = jFileChooser.getSelectedFile().getAbsolutePath
());

        if (outputPath == null) {
            return;
        }
        new Thread(new Runnable() {

            @Override
            public void run() {
                count.set(0);
                progressBar.setIndeterminate(false);
                progressBar.setMinimum(0);
                progressBar.setMaximum(urlvector.size());
                progressBar.setValue(0);
                progressBar.setStringPainted(true);
            }
        }).start();
    }
});

```

```

HashMap<Integer, Vector<String>> map = new HashMap<
>());

int pronum = 3;
for (int i = 0; i < pronum; i++) {
    map.put(i, new Vector<>());
}
for (int i = 0; i < urlvector.size(); i++) {
    int j = i % pronum;
    map.get(j).add(urlvector.elementAt(i));
}
for (int i = 0; i < pronum; i++) {
    if (map.get(i).size() != 0) {
        new scan(map.get(i), self);
    }
}
int c = 0;
AtomicInteger c1 = new AtomicInteger(0);
new Thread(new Runnable() {
    @Override
    public void run() {
        int c2;
        do {
            c2 = c1.get();
            Dimension dimension = jProgressBar.getSi
ize();

            Rectangle rectangle = new Rectangle(0,
0, dimension.width, dimension.height);
            jProgressBar.setValue(c2);
            jProgressBar.paintImmediately(rectangle

);

            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } while (c2 != urlvector.size());
    }
}).start();
while (true) {
    c = count.get();
    c1.set(c);
    if (c == urlvector.size()) {

```

```

        JOptionPane.showMessageDialog(self, "所有
URL 过滤完成", "过滤完成", JOptionPane.PLAIN_MESSAGE);
        jButton5.setEnabled(true);
        break;
    }
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e1) {

    }
}
}).start();
}

class scan implements Runnable {
    Vector<String> vector = null;
    normalFilter nFilter = null;

    public scan(Vector<String> v, normalFilter nf) {
        vector = v;
        nFilter = nf;
        new Thread(this).start();
    }

    @Override
    public void run() {
        for (String string : vector) {
            BufferedWriter bWriter = null;
            try {
                Pattern pattern = Pattern.compile("\\w+");
                Matcher matcher = pattern.matcher(string);
                String filename = "";
                while (matcher.find()) {
                    filename += matcher.group();
                }
                File file = new File(outputPath + "/" + filename
e + ".txt");

                if (!file.exists()) {
                    file.createNewFile();
                }
                bWriter = new BufferedWriter(new OutputStreamWr
iter(new FileOutputStream(file)));
                String source = filter.getSource(string);

```



```

        String regEx_script = "<script[^>]*?>[\\s\\S]*?<\\/script>"; // 定义 script 的正则表达式
        String regEx_style = "<style[^>]*?>[\\s\\S]*?<\\/style>"; // 定义 style 的正则表达式
        String regEx_html = "<[^>]+>"; // 定义 HTML 标签的正则表达式
        String regEx_zhuanyi = "&[\\s\\S]*?"; // 定义转义字符&copy;等的正则表达式
        String regEx_space = "\\s{2,}|\\t";

        Pattern p_script = Pattern.compile(regEx_script, Pattern.CASE_INSENSITIVE);
        Matcher m_script = p_script.matcher(source);
        source = m_script.replaceAll(""); // 过滤 script 标签

        Pattern p_style = Pattern.compile(regEx_style, Pattern.CASE_INSENSITIVE);
        Matcher m_style = p_style.matcher(source);
        source = m_style.replaceAll(""); // 过滤 style 标签

        Pattern p_html = Pattern.compile(regEx_html, Pattern.CASE_INSENSITIVE);
        Matcher m_html = p_html.matcher(source);
        source = m_html.replaceAll(""); // 过滤 html 标签

        Pattern p_zhuanyi = Pattern.compile(regEx_zhuanyi, Pattern.CASE_INSENSITIVE);
        Matcher m_zhuanyi = p_zhuanyi.matcher(source);
        source = m_zhuanyi.replaceAll(""); // 过滤转义字符

        Pattern p_space = Pattern.compile(regEx_space, Pattern.CASE_INSENSITIVE);
        Matcher m_space = p_space.matcher(source);
        source = m_space.replaceAll("\\n");

        return source.trim(); // 返回文本字符串
    }

    public static void highLight(JTextArea jTextArea, Vector<String> vector) {
        String text = jTextArea.getText();
        int pos = 0;
        Highlighter highlighter = jTextArea.getHighlighter();
        DefaultHighlightPainter painter = new DefaultHighlightPainter(Color.green);
    }

```

```

        for (String string : vector) {
            while ((pos = text.indexOf(string, pos)) >= 0) {
                try {
                    highlighter.addHighlight(pos, pos + string.length(), pa
inter);

                    pos += string.length();
                } catch (Exception e) {

                }
            }
        }
    }
}

```

```

    public static Vector<String> readWords(String fileName) throws Exceptio
n {
        Vector<String> vector = new Vector<>();
        BufferedReader bReader = new BufferedReader(
            new InputStreamReader(new FileInputStream(new File(fileName
)), "GBK"));
        String line;
        while ((line = bReader.readLine()) != null) {
            vector.add(line);
        }
        return vector;
    }
}

```

```

    public static Vector<String> importLib(Vector<JButton> bVector, JLabel
jLabel, normalFilter self) {
        Vector<String> vector = new Vector<>();
        JFileChooser jFileChooser = new JFileChooser();
        jFileChooser.setCurrentDirectory(new File("."));
        jFileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
        jFileChooser.setAcceptAllFileFilterUsed(false);
        jFileChooser.addChoosableFileFilter(new FileFilter() {

```

```

            @Override
            public String getDescription() {
                // TODO Auto-generated method stub
                return "文本文件(*.txt)";
            }

```

```

            @Override
            public boolean accept(File f) {
                // TODO Auto-generated method stub

```



```

        return f.getName().endsWith(".txt") || f.isDirectory();
    }
});
jFileChooser.showDialog(new JLabel(), "选择");
try {
    String fileName = jFileChooser.getSelectedFile().getAbsolutePath();
    h();

    vector = filter.readWords(fileName);
    jLabel.setText("已导入目标库: " + fileName);
    for (JButton jButton : bVector) {
        jButton.setEnabled(true);
    }
} catch (Exception e2) {
    new JOptionPane().showMessageDialog(self, "目标库文件错误", "ERROR", JOptionPane.ERROR_MESSAGE);
    jLabel.setText("请导入库文件");
    for (JButton jButton : bVector) {
        jButton.setEnabled(false);
    }
}
return vector;
}
}

```

8.2.3 URLfilter.java

```

package filter;

import java.io.InputStream;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.Console;

```

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.lang.reflect.Field;
import java.net.URL;
import java.text.NumberFormat;
import java.util.HashMap;
import java.util.Iterator;
import java.util.TimerTask;
import java.util.Vector;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.naming.spi.DirStateFactory.Result;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.text.JTextComponent;
import javax.swing.text.DefaultHighlighter.DefaultHighlightPainter;
import javax.swing.text.Highlighter;
import javax.swing.text.Highlighter.Highlight;
import javax.swing.text.Highlighter.HighlightPainter;

public class URLfilter extends JFrame {
    public static void main(String[] args) {
        new normalFilter();
    }
}

class normalFilter extends JFrame {
    JPanel mainPanel = new JPanel();
    JPanel buttonPanel = new JPanel();

```

```

JButton jButton1 = new JButton("导入敏感词库");
JButton jButton2 = new JButton("高亮显示敏感词");
JButton jButton3 = new JButton("爬取目标 URL");
JButton jButton4 = new JButton("导入目标 URL 库");
JButton jButton5 = new JButton("开始批量爬取");
JLabel jLabel = new JLabel("请导入敏感词库");
JLabel urLabel = new JLabel("请导入目标 URL 库");
JTextArea jTextArea = new JTextArea();
JScrollPane jScrollPane = new JScrollPane(jTextArea);
JProgressBar jProgressBar = new JProgressBar();
normalFilter self = this;
String fileName = null;
String source = null;
String outputPath = null;
Vector<String> urlvector = null;
Vector<String> wordvector = null;
AtomicInteger count = new AtomicInteger(0);

public normalFilter() {
    this.setTitle("URL Filter");
    this.setSize(800, 600);
    this.setLocation(200, 200);
    this.setVisible(true);
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.add(mainPanel);
    mainPanel.setLayout(new BorderLayout());
    mainPanel.add(jScrollPane, BorderLayout.CENTER);
    mainPanel.add(buttonPanel, BorderLayout.SOUTH);
    buttonPanel.setLayout(new GridLayout(4, 2));
    buttonPanel.add(jButton3);
    buttonPanel.add(jButton2);
    buttonPanel.add(jButton1);
    buttonPanel.add(jLabel);
    buttonPanel.add(jButton4);
    buttonPanel.add(urLabel);
    buttonPanel.add(jButton5);
    buttonPanel.add(jProgressBar);
    jTextArea.setLineWrap(true);
    jTextArea.setWrapStyleWord(true);
    jTextArea.setEditable(false);
    jButton2.setEnabled(false);
    jButton4.setEnabled(false);
    jButton5.setEnabled(false);
}

```

```

jButton1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Vector<JButton> bVector = new Vector<>();
        bVector.add(jButton2);
        bVector.add(jButton4);
        wordvector = filter.importLib(bVector, jLabel, self);
        jButton5.setEnabled(false);
    }
});

jButton2.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        filter.highLight(jTextArea, wordvector);
    }
});

jButton3.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        String source = JOptionPane.showInputDialog(self, "请输入目
标 URL");

        // 获取 url
        jButton3.setEnabled(false);
        new Thread(new Runnable() {

            @Override
            public void run() {
                try {
                    try {
                        // 创建 URL 对象，将网络资源路径传递到该对象进行
                        绑定
                        URL url2 = new URL(source);
                        // 通过 url1 对象打开并且激活网络流来获取网页资源
                        InputStream openStream = url2.openStream();
                        InputStreamReader read = new InputStreamRea
                        der(openStream, "UTF-8");

                        BufferedReader br = new BufferedReader(read
                        );

                        // 创建爬虫后要写入的文件

```

```

        FileOutputStream fos = new FileOutputStream
(new File("F:\\study data\\大二\\Java\\" +
        "idea\\javurl\\output\\html\\url.tx
t"));

        OutputStreamWriter osw = new OutputStreamWr
iter(fos);

        BufferedWriter bw = new BufferedWriter(osw)
;

        // 定义空变量来储存爬下来的内容
        String str = "";
        while ((str = br.readLine()) != null) {
            bw.write(str);
            bw.newLine();
            bw.flush();
        }
        // 关闭资源
        bw.close();
        osw.close();
        fos.close();
        br.close();
        read.close();
        openStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }

    jProgressBar.setIndeterminate(true);
    jProgressBar.setStringPainted(false);
    jTextArea.setText("");
    jTextArea.append(filter.deleteHTMLTag(filter.ge
tSource(source)));

    JOptionPane.showMessageDialog(self, source + "
爬取完成", "网页爬取完成", JOptionPane.PLAIN_MESSAGE);
    } catch (Exception e2) {
        JOptionPane.showMessageDialog(self, "目标 URL 异
常", "无法提取网页信息", JOptionPane.ERROR_MESSAGE);
        jButton2.setEnabled(false);
    }finally {
        jProgressBar.setIndeterminate(false);
        jProgressBar.setMaximum(1);
        jProgressBar.setValue(0);
    }

```

```

        progressBar.setBorderPainted(false);
        jButton3.setEnabled(true);
    }
}
}).start();
}
});

jButton4.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Vector<JButton> bVector = new Vector<>();
        bVector.add(jButton5);
        urlvector = filter.importLib(bVector, urLabel, self);
    }
});

jButton5.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        jButton5.setEnabled(false);
        JFileChooser jFileChooser = new JFileChooser();
        jFileChooser.setCurrentDirectory(new File("."));
        jFileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_
ONLY);

        jFileChooser.showDialog(new Label(), "选择输出文件夹");
        outputPath = jFileChooser.getSelectedFile().getAbsolutePath
();

        if (outputPath == null) {
            return;
        }
        new Thread(new Runnable() {

            @Override
            public void run() {
                count.set(0);
                progressBar.setIndeterminate(false);
                progressBar.setMinimum(0);
                progressBar.setMaximum(urlvector.size());
                progressBar.setValue(0);
                progressBar.setStringPainted(true);
                HashMap<Integer, Vector<String>> map = new HashMap<
>();

                int pronum = 3;

```

```

    for (int i = 0; i < pronum; i++) {
        map.put(i, new Vector<>());
    }
    for (int i = 0; i < urlvector.size(); i++) {
        int j = i % pronum;
        map.get(j).add(urlvector.elementAt(i));
    }
    for (int i = 0; i < pronum; i++) {
        if (map.get(i).size() != 0) {
            new scan(map.get(i), self);
        }
    }
    int c = 0;
    AtomicInteger c1 = new AtomicInteger(0);
    new Thread(new Runnable() {
        @Override
        public void run() {
            int c2;
            do {
                c2 = c1.get();
                Dimension dimension = jProgressBar.getSi
ize();

                Rectangle rectangle = new Rectangle(0,
0, dimension.width, dimension.height);
                jProgressBar.setValue(c2);
                jProgressBar.paintImmediately(rectangle

);

                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            } while (c2 != urlvector.size());
        }
    }).start();
    while (true) {
        c = count.get();
        c1.set(c);
        if (c == urlvector.size()) {
            JOptionPane.showMessageDialog(self, "所有
URL 过滤完成", "过滤完成", JOptionPane.PLAIN_MESSAGE);
            jButton5.setEnabled(true);
            break;

```

```

    }
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e1) {

    }
}
}
}).start();
}

class scan implements Runnable {
    Vector<String> vector = null;
    normalFilter nFilter = null;

    public scan(Vector<String> v, normalFilter nf) {
        vector = v;
        nFilter = nf;
        new Thread(this).start();
    }

    @Override
    public void run() {
        for (String string : vector) {
            BufferedWriter bWriter = null;
            try {
                Pattern pattern = Pattern.compile("\\w+");
                Matcher matcher = pattern.matcher(string);
                String filename = "";
                while (matcher.find()) {
                    filename += matcher.group();
                }
                File file = new File(outputPath + "/" + filename
e + ".txt");

                if (!file.exists()) {
                    file.createNewFile();
                }
                bWriter = new BufferedWriter(new OutputStreamWr
iter(new FileOutputStream(file)));
                String source = filter.getSource(string);
                source = filter.deleteHTMLTag(source);
                bWriter.write(source);
                bWriter.newLine();
                for (String word : wordvector) {

```



```

        String regEx_zhuanyi = "&[\\s\\S]*?"; // 定义转义字符&copyys;等的正则
表达式
        String regEx_space = "\\s{2,}|\t";

        Pattern p_script = Pattern.compile(regEx_script, Pattern.CASE_INSENSITIVE);
        Matcher m_script = p_script.matcher(source);
        source = m_script.replaceAll(""); // 过滤 script 标签

        Pattern p_style = Pattern.compile(regEx_style, Pattern.CASE_INSENSITIVE);
        Matcher m_style = p_style.matcher(source);
        source = m_style.replaceAll(""); // 过滤 style 标签

        Pattern p_html = Pattern.compile(regEx_html, Pattern.CASE_INSENSITIVE);
        Matcher m_html = p_html.matcher(source);
        source = m_html.replaceAll(""); // 过滤 html 标签

        Pattern p_zhuanyi = Pattern.compile(regEx_zhuanyi, Pattern.CASE_INSENSITIVE);
        Matcher m_zhuanyi = p_zhuanyi.matcher(source);
        source = m_zhuanyi.replaceAll(""); // 过滤转义字符

        Pattern p_space = Pattern.compile(regEx_space, Pattern.CASE_INSENSITIVE);
        Matcher m_space = p_space.matcher(source);
        source = m_space.replaceAll("\n");

        return source.trim(); // 返回文本字符串
    }

    public static void highLight(JTextArea jTextArea, Vector<String> vector) {
        String text = jTextArea.getText();
        int pos = 0;
        Highlighter highlighter = jTextArea.getHighlighter();
        DefaultHighlightPainter painter = new DefaultHighlightPainter(Color.green);
        for (String string : vector) {
            while ((pos = text.indexOf(string, pos)) >= 0) {
                try {
                    highlighter.addHighlight(pos, pos + string.length(), painter);
                } catch (Exception e) {
                    // 忽略异常
                }
            }
        }
    }

```

```

        pos += string.length();
    } catch (Exception e) {

    }
}
}
}

    public static Vector<String> readWords(String fileName) throws Exception
n {
    Vector<String> vector = new Vector<>();
    BufferedReader bReader = new BufferedReader(
        new InputStreamReader(new FileInputStream(new File(fileName
    )), "GBK"));
    String line;
    while ((line = bReader.readLine()) != null) {
        vector.add(line);
    }
    return vector;
}

    public static Vector<String> importLib(Vector<JButton> bVector, JLabel
jLabel, normalFilter self) {
    Vector<String> vector = new Vector<>();
    JFileChooser jFileChooser = new JFileChooser();
    jFileChooser.setCurrentDirectory(new File("."));
    jFileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
    jFileChooser.setAcceptAllFileFilterUsed(false);
    jFileChooser.addChoosableFileFilter(new FileFilter() {

        @Override
        public String getDescription() {
            // TODO Auto-generated method stub
            return "文本文件(*.txt)";
        }

        @Override
        public boolean accept(File f) {
            // TODO Auto-generated method stub
            return f.getName().endsWith("txt") || f.isDirectory();
        }
    });
    jFileChooser.showDialog(new Label(), "选择");
    try {

```

```

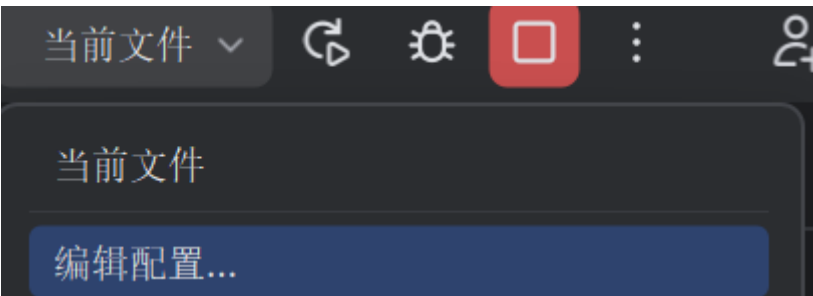
        String fileName = jFileChooser.getSelectedFile().getAbsolutePath();
    h());

    vector = filter.readWords(fileName);
    jLabel.setText("已导入目标库: " + fileName);
    for (JButton jButton : bVector) {
        jButton.setEnabled(true);
    }
} catch (Exception e2) {
    new JOptionPane().showMessageDialog(self, "目标库文件错误
", "ERROR", JOptionPane.ERROR_MESSAGE);
    jLabel.setText("请导入库文件");
    for (JButton jButton : bVector) {
        jButton.setEnabled(false);
    }
}
return vector;
}
}

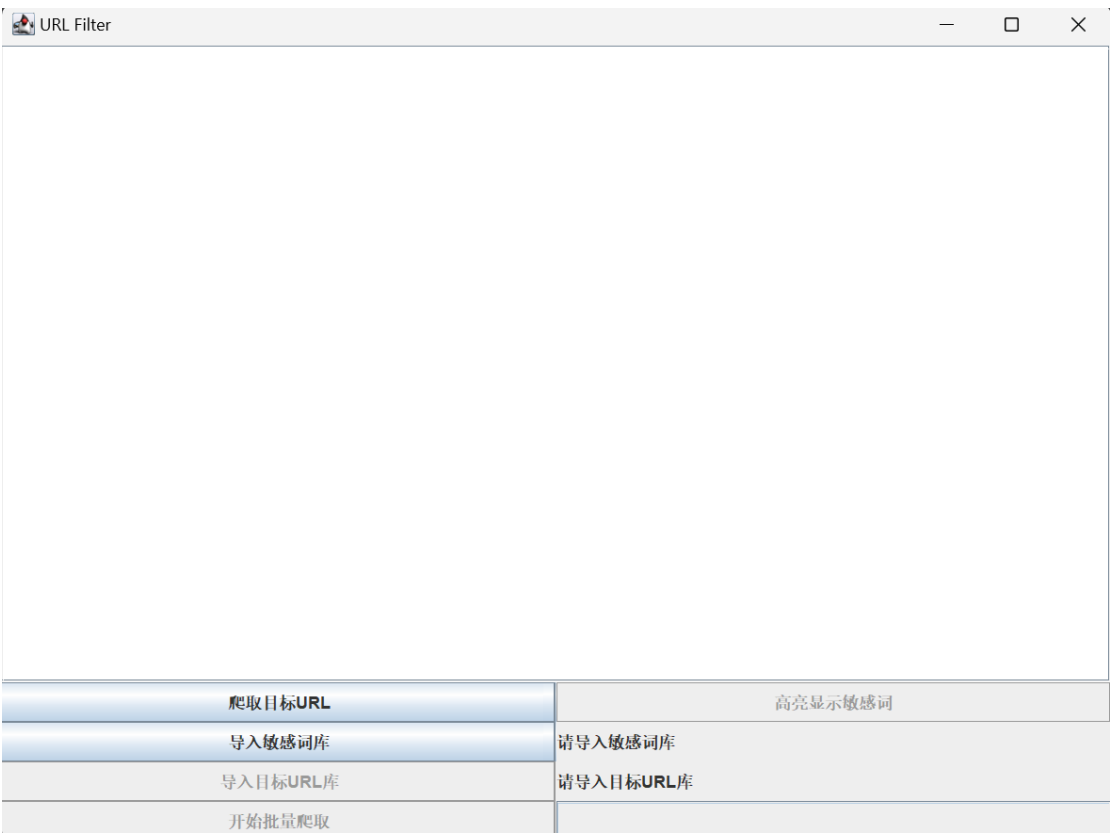
```

九、爬取并分析网页敏感词程序运行结果与测试分析

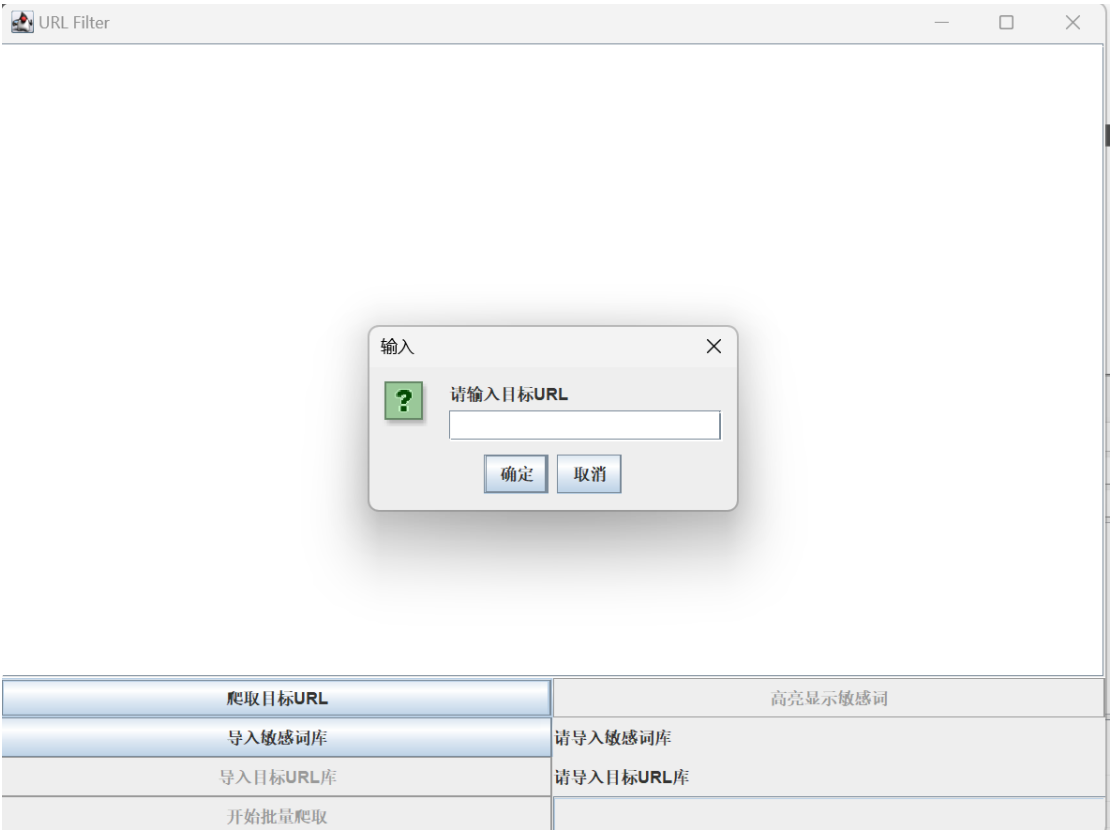
9.1 运行代码



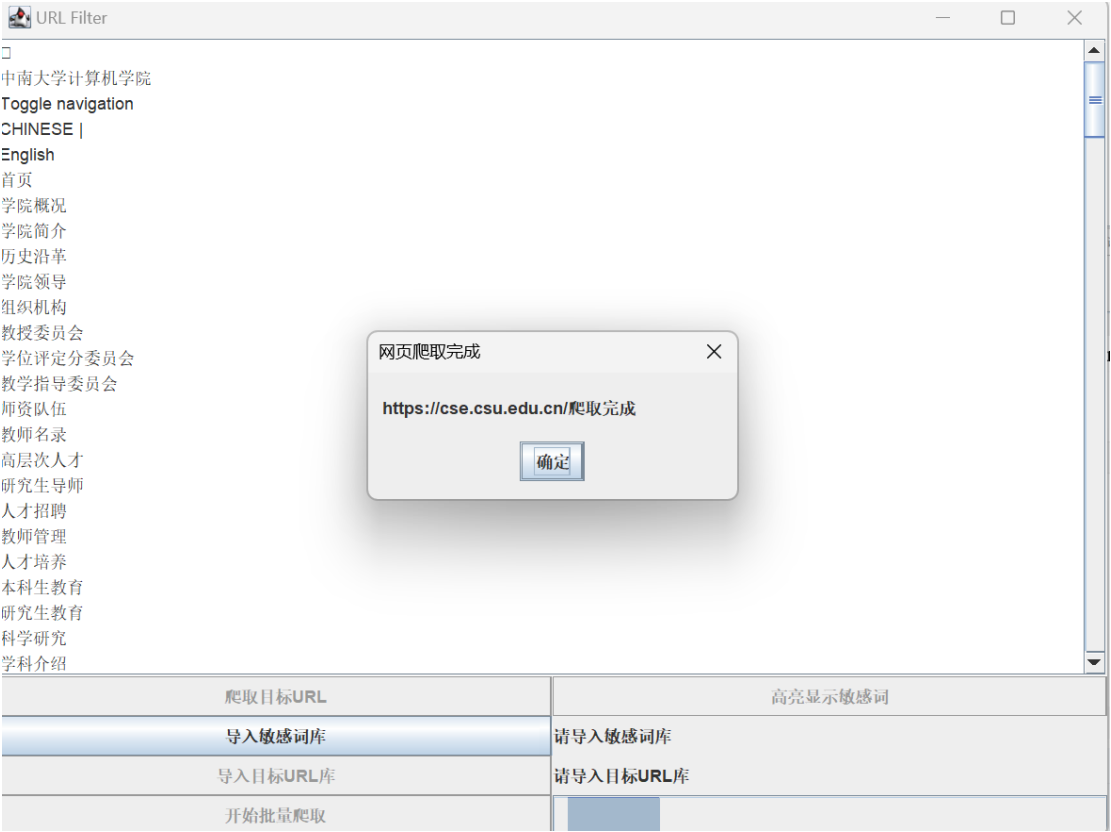
9.2 主界面



9.3 点击爬取目标 url



输入[中南大学计算机学院 \(csu.edu.cn\)](https://cse.csu.edu.cn/): <https://cse.csu.edu.cn/>

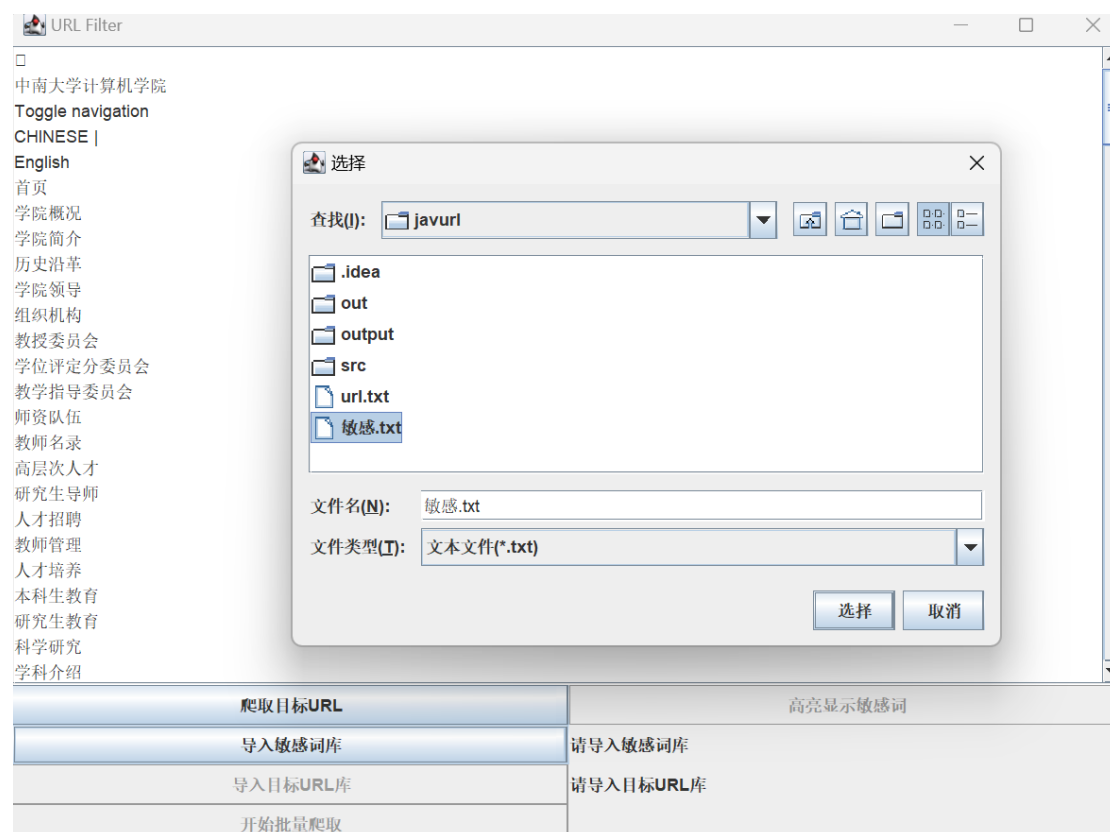


爬取目标 http

```
敏感.txt  txt.txt  https://blogs.cdn.net/q36761831/article/details/81490380.txt  https://v.qq.com  output\...\url.txt x 265 ^ v
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>中南大学计算机学院</title><META Name="keywords" Content="中南大学计算机学院" />
8
9      <link href="css/bootstrap.min.css" rel="stylesheet">
10     <link rel="stylesheet" type="text/css" href="css/main.css" />
11     <!-- Hoshizora Rin ga ka wa i i -->
12     <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
13     <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
14     <!--[if lt IE 9]>
15
16     <![endif]-->
17     <!--Announced by Visual SiteBuilder 9-->
18     <link rel="stylesheet" type="text/css" href="_sitegray/_sitegray_d.css" />
19     <script language="javascript" src="_sitegray/_sitegray.js"></script>
20     <!-- CustomerNO:7765626265723230747a475c52505742000000004055 -->
21     <link rel="stylesheet" type="text/css" href="index.vsb.css" />
22     <script type="text/javascript" src="/system/resource/js/counter.js"></script>
23     <script type="text/javascript">_jsq_(1001,'/index.jsp',-1,1556422056)</script>
24 </head>
25 <body>
26     <!--头部-->
27     <div class="zn-header">
28         <nav class="navbar">
29             <!--logo-->
30             <div class="container">
31                 <div class="navbar-header">
```

9.4 点击导入敏感词库

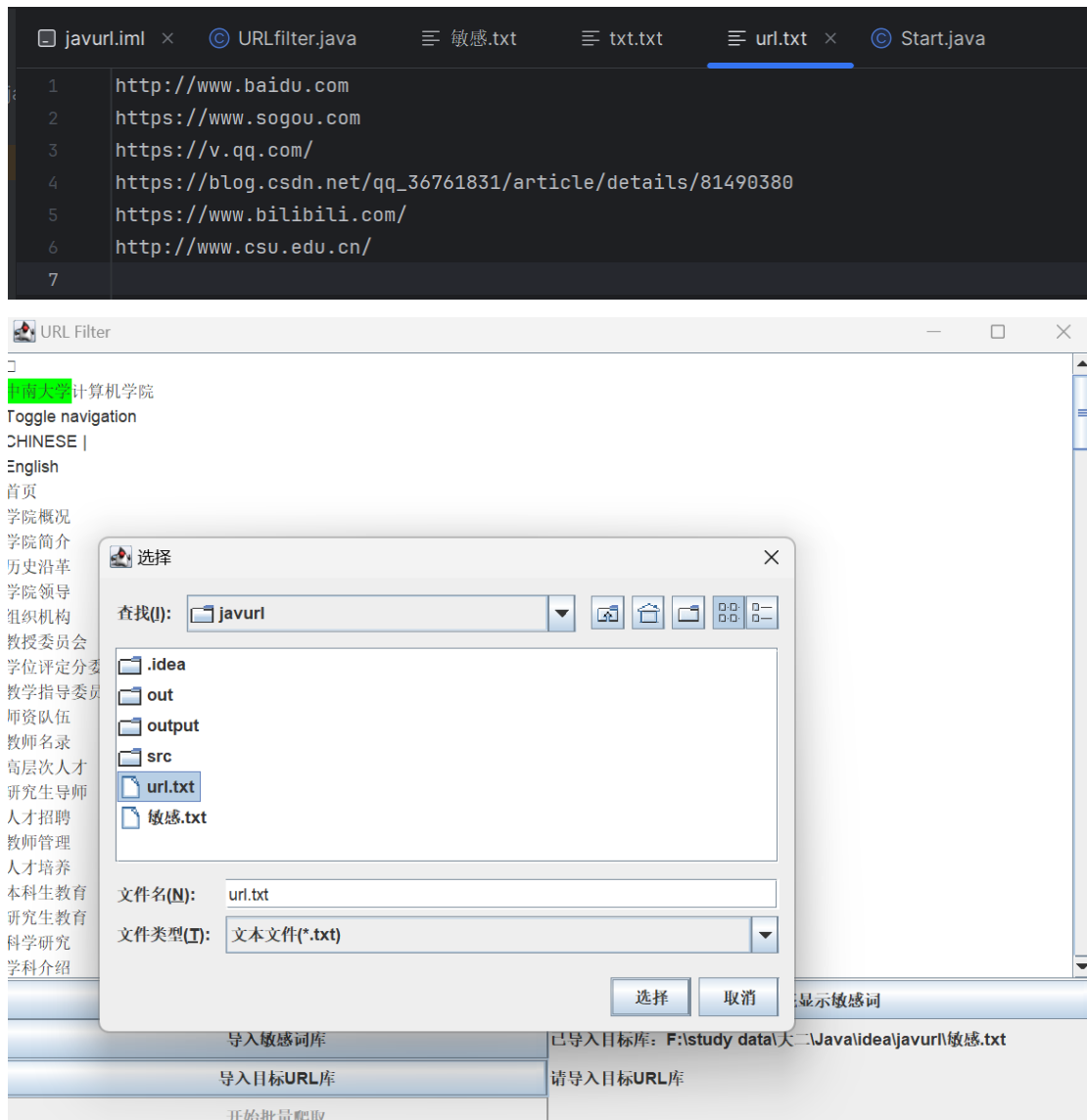
```
javurl.iml  URLfilter.java  敏感.txt x  txt.txt  url.txt  Start.java
1  蔡徐坤
2  中南大学
3  我
```



9.5 点击高亮显示

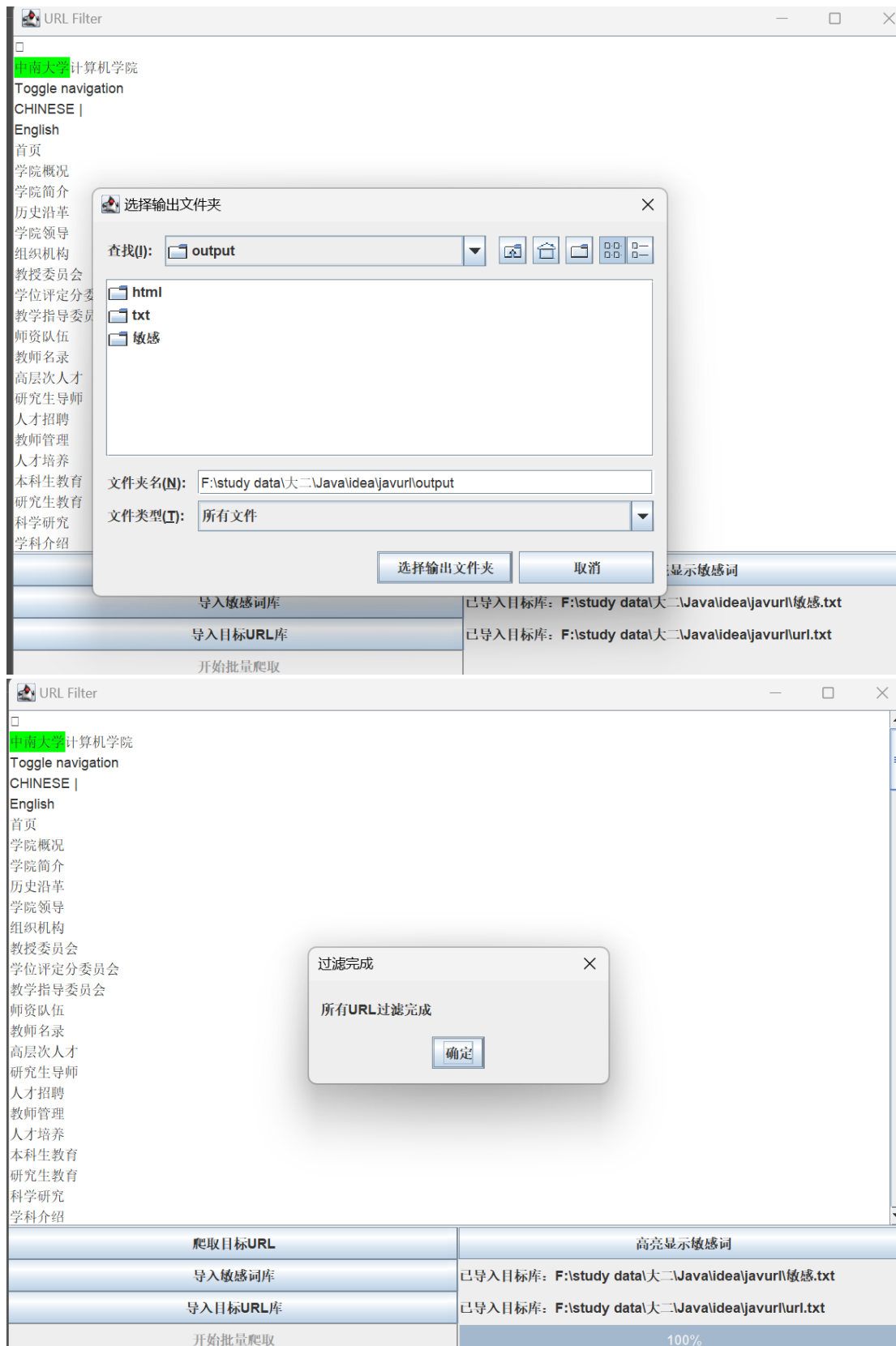


9.6 点击导入目标 url 库

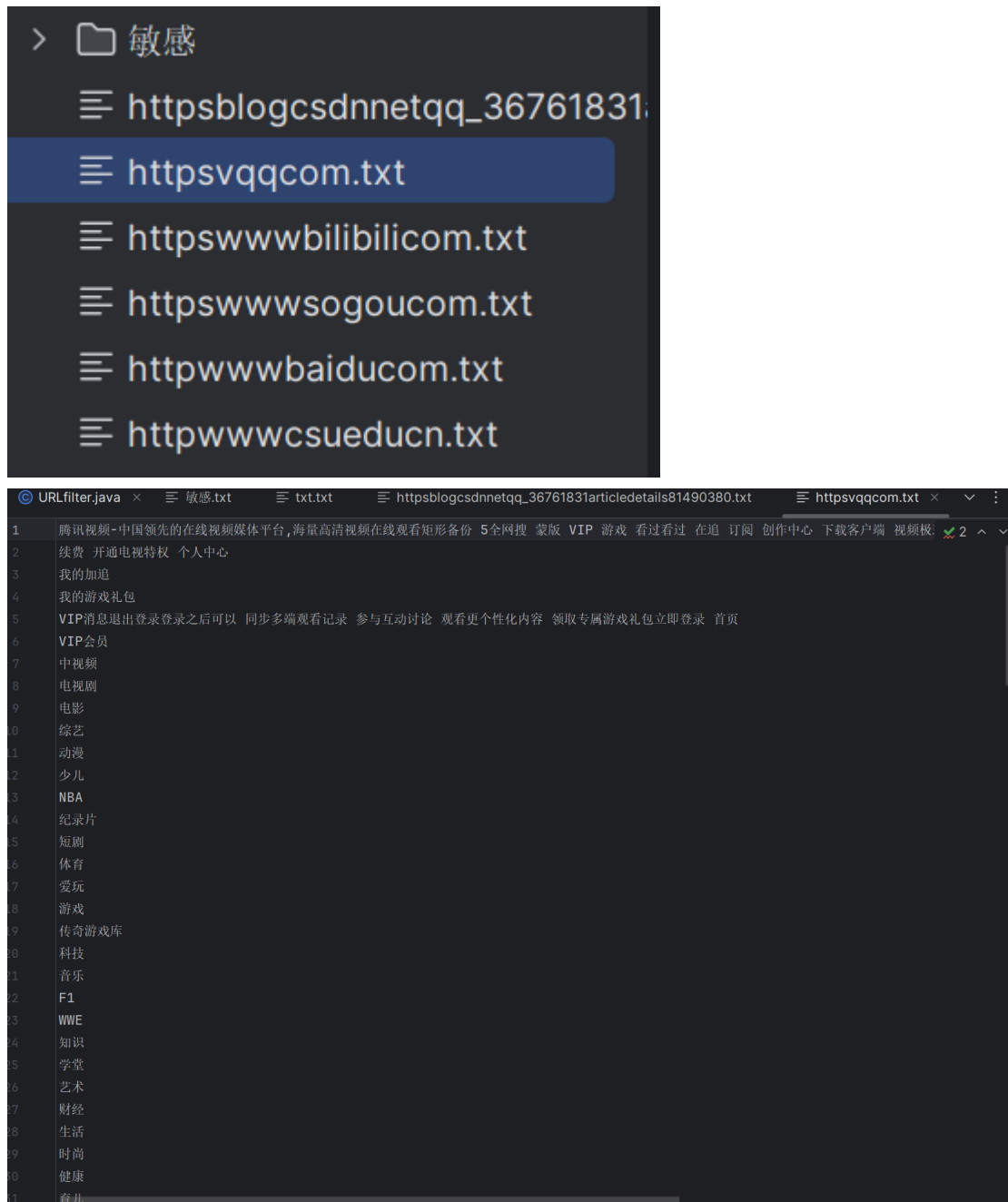


9.7 点击开始批量爬取

选择输出文件夹



9.8 查看爬取内容和敏感词



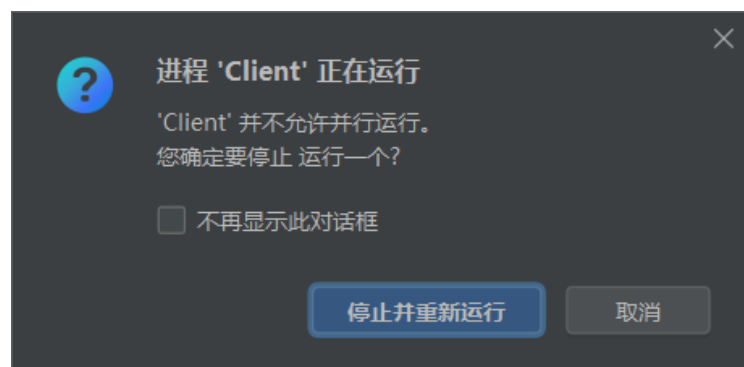
十、结论与心得

10.1 网络聊天程序

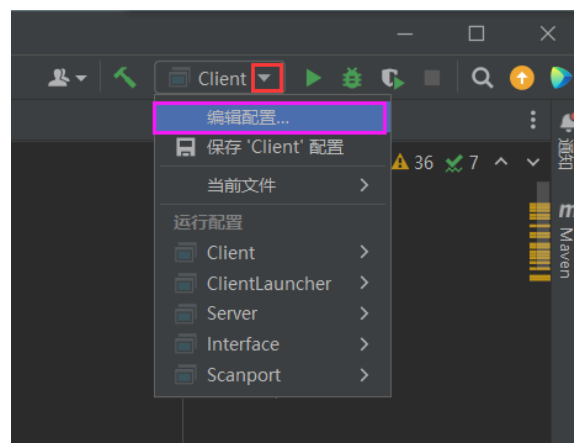
10.1.1 思考

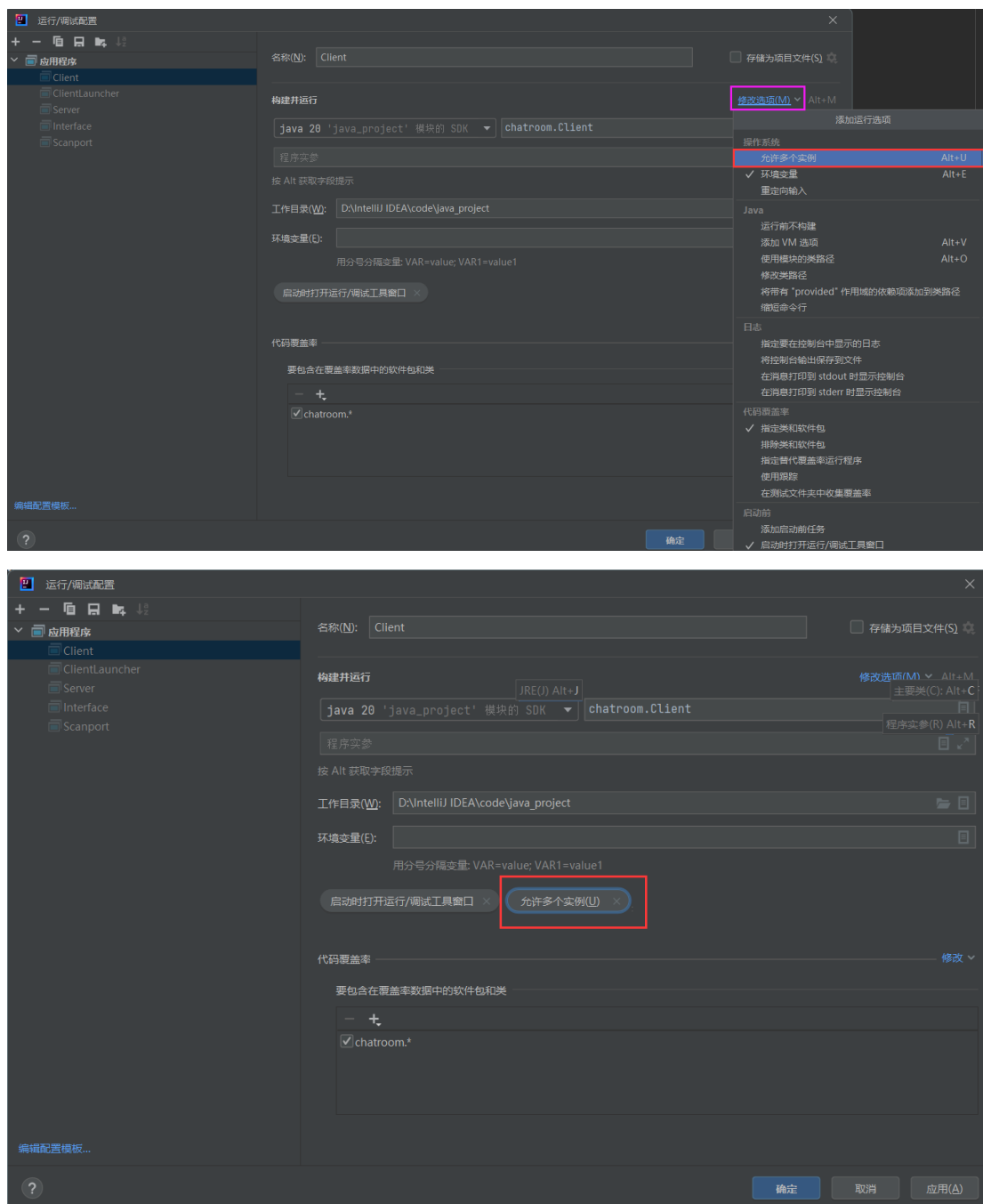
IDEA 中无法运行多线程的解决方法

该实验中，客户端需要实现多线程，但是第二次运行客户端程序时会跳出下列提示：



那么这个问题如何解决呢？





最后点击确认就好。

10.1.2 心得

通过使用 Java Socket 编程开发聊天室的实验，我深入理解了网络编程的基本概念和原理，并学会了如何处理多线程、错误和异常等情况。同时，我也认识到了设计良好的客户端和服务端架构的重要性，以及安全性和权限控制的需求。这次实验对于我提升网络编程技能和软件设计能力有很大的帮助。

10.1.3 结论

实现聊天室需要对网络编程有一定的了解。这包括理解 TCP/IP 协议、Socket 编程概念和网络通信原理等。在实验中，我们使用 Socket 类来建立客户端和服务端之间的连接，以及利用输入输出流进行数据的传输。

多线程编程：聊天室需要同时处理多个客户端的连接和消息交互。因此，多线程编程是不可或缺的。在实验中，我们需要为每个客户端连接创建一个独立的线程，并实现线程之间的同步和通信。这样可以使得每个客户端的交互是独立的，互不干扰。

10.2 爬取并分析网页敏感词程序

10.2.1 思考

网页爬取：使用 Java 的 URL 类库可以轻松地获取网页的 HTML 源代码，但需要注意网络连接的异常处理和数据流的释放，以确保程序的稳定性和资源的合理利用。

敏感词识别：为了实现敏感词的识别，可以使用字符串匹配算法（如 KMP 算法）或正则表达式来检测敏感词是否存在于提取到的文本中。同时，敏感词库的维护和更新也需要被考虑。

心得：

结论：

10.2.2 心得

网页爬取和分析是一项有趣而具有挑战性的任务。通过编写 Java URL 编程实现，我深入了解了网络数据的获取和处理过程，提高了对网络编程的理解。敏感词分析是一项重要的文本处理任务，尤其对于信息安全和内容过滤方面非常有用。通过实现敏感词的提取和高亮显示，我体会到了文本处理技术在实际应用中的价值和作用。

10.2.3 结论

Java URL 编程提供了强大的工具和类库，可以方便地实现网页爬取和分析任务。

网页爬取需要处理网络连接异常和资源释放，保证程序的稳定性和效率。敏感词的识别可以使用字符串匹配算法或正则表达式，同时需要维护和更新敏感词库。

网页敏感词分析可以帮助实现信息安全和内容过滤等功能，对于网络数据处理和保护具有重要意义。

