

```

1 package SMTP;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6 import java.io.PrintWriter;
7 import java.net.InetAddress;
8 import java.net.Socket;
9 import java.net.UnknownHostException;
10
11 import sun.misc.BASE64Encoder;
12
13 public class SMTPClient {
14     String mailContent = null;           // 邮件正文
15     String mailServer = null;            // 发送者邮件服务器域名
16     int port = -1;                        // 邮件服务器端口
17     String mailSender = null;            // 发送者邮箱
18     String mailReceiver = null;          // 邮件接收者信箱
19     Socket socket = null;                // Socket对象
20     PrintWriter outToServer = null;      // 向SMTP服务器发送报文的输出流
21     BufferedReader inFromServer = null;  // 从SMTP服务器接收响应的输入流
22     String userName = null;              // 邮件发送者的账号
23     String pwd = null;                   // 邮件发送者的账号密码
24
25     public SMTPClient(String mailServer, int port, String userName, String pwd) {
26         this.mailServer = mailServer;
27         this.port = port;
28         this.userName = userName;
29         this.pwd = pwd;
30     }
31     /**
32      * @return the mailContent
33      */
34     public String getMailContent() {
35         return mailContent;
36     }
37     /**
38      * @param mailContent the mailContent to set
39      */
40     public void setMailContent(String mailContent) {
41         this.mailContent = mailContent;
42     }
43     /**
44      * @return the mailServer
45      */
46     public String getMailServer() {
47         return mailServer;
48     }
49     /**
50      * @param mailServer the mailServer to set
51      */
52     public void setMailServer(String mailServer) {
53         this.mailServer = mailServer;
54     }
55     /**
56      * @return the port
57      */
58     public int getPort() {
59         return port;
60     }
61     /**
62      * @param port the port to set
63      */
64     public void setPort(int port) {
65         this.port = port;
66     }
67     /**
68      * @return the mailSender
69      */
70     public String getMailSender() {
71         return mailSender;

```

```

72     }
73     /**
74      * @param mailSender the mailSender to set
75      */
76     public void setMailSender(String mailSender) {
77         this.mailSender = mailSender;
78     }
79     /**
80      * @return the mailReceiver
81      */
82     public String getMailReceiver() {
83         return mailReceiver;
84     }
85     /**
86      * @param mailReceiver the mailReceiver to set
87      */
88     public void setMailReceiver(String mailReceiver) {
89         this.mailReceiver = mailReceiver;
90     }
91
92     boolean isConnected(){
93         if(this.socket == null) return false;
94         else return this.socket.isConnected();
95     }
96     public void connect(){
97         if(!isConnected()){
98             try {
99                 this.socket = new Socket(mailServer,port); //创建Socket对象并连接到指定的域名和端口
100                //将SOCKET输入流连接到带缓冲功能的输入流BufferedReader, 以便一次读一行来自服务器的应答报文
101                this.inFromServer =
102                    new BufferedReader(new InputStreamReader(this.socket.getInputStream()));
103                //将SOCKET输出流连接到带缓冲功能的输出流PrintWriter, 以便一次输出一行报文到服务器
104                this.outToServer = new PrintWriter(this.socket.getOutputStream(),true);
105            } catch (UnknownHostException e) {
106                e.printStackTrace();
107            } catch (IOException e) {
108                e.printStackTrace();
109            }
110        }
111    }
112    public void disconnect(){
113        if(isConnected()){
114            try {
115                this.inFromServer.close();
116                this.outToServer.close();
117                this.socket.close();
118            } catch (IOException e) {
119                e.printStackTrace();
120            }
121        }
122    }
123    /**
124     * 从服务器读入一行应答, 并打印显示
125     */
126    public void getResponse(){
127        try {
128            String response = this.inFromServer.readLine(); //从服务器读入一行应答
129            System.out.println("MAILServer: " + response); //打印显示
130        } catch (IOException e) {
131            e.printStackTrace();
132        }
133    }
134    /**
135     * 向服务器发送命令, 并打印显示
136     * @param cmd: 命令
137     */
138    public void sendCommand(String cmd){
139        this.outToServer.println(cmd); //向服务器发送命令
140        System.out.println("Client: " + cmd); //打印显示
141    }
142

```

```

143 public void sendMail(){
144     try {
145         getResponse();          //读取来自服务器的第一行应答，并显示在屏幕上
146
147         //将用户的帐号和密码以BASE64格式进行编码以便进行服务器身份验证
148         BASE64Encoder encoder = new BASE64Encoder();
149         String encodedUserName = encoder.encode(this.userName.getBytes());
150         String encodedPwd = encoder.encode(this.pwd.getBytes());
151
152         //和服务器会话，发送EHLO hostname命令
153         String hostName = InetAddress.getLocalHost().getHostName(); //得到本机主机名
154         sendCommand("EHLO " + hostName); //发送EHLO hostname命令到服务器
155         //读入来自服务器的5行应答，并显示在屏幕上
156         for(int i = 0; i < 5; i++){
157             getResponse();
158         }
159
160         //和服务器会话，发送AUTH LOGIN命令，请求身份验证
161         sendCommand("AUTH LOGIN"); //向服务器发送AUTH LOGIN命令
162         //读入来自服务器的应答，并显示在屏幕上
163         getResponse();
164
165         //向服务器发送自己的经过BASE64编码的帐号
166         sendCommand(encodedUserName); //向服务器发送经过BASE64编码的帐号
167         //读入来自服务器的应答，并显示在屏幕上
168         getResponse();
169
170         //向服务器发送自己的密码
171         sendCommand(encodedPwd); //向服务器发送经过BASE64编码的密码
172         //读入来自服务器的应答，并显示在屏幕上
173         getResponse();
174
175         //向服务器发送MAIL FROM: 发件人地址
176         sendCommand("MAIL FROM: <" + this.mailSender + ">");//向服务器发送MAIL FROM: 发件人地址
177         //读入来自服务器的应答，并显示在屏幕上
178         getResponse();
179
180         //向服务器发送RCPT TO: 收件人地址
181         sendCommand("RCPT TO: <" + this.mailReceiver + ">");//向服务器发送RCPT TO: 收件人地址
182         //读入来自服务器的应答，并显示在屏幕上
183         getResponse();
184
185         //请求发送邮件正文
186         sendCommand("DATA"); //向服务器发送"DATA"命令，请求发送邮件正文
187         //读入来自服务器的应答，并显示在屏幕上
188         getResponse();
189
190         //开始发送邮件正文
191         sendCommand(mailContent);
192         //发送邮件结束标志
193         sendCommand(".");
194         sendCommand("QUIT");
195         //读入来自服务器的应答，并显示在屏幕上
196         getResponse();
197
198     } catch (IOException e) {
199         e.printStackTrace();
200     }
201 }
202
203 public static void main(String[] args) {
204     SMTPClient smtpClient =
205         new SMTPClient("smtp.163.com",25,"hustnetwork","network123");
206     smtpClient.setMailSender("hustnetwork@163.com");
207     smtpClient.setMailReceiver("hustnetwork@163.com");
208     smtpClient.setMailContent("From: " + smtpClient.getMailSender() + "\n" +
209                             "To: " + smtpClient.getMailReceiver() + "\n" +
210                             "Subject: " + "Test" + "\n\n" +
211                             "Hi\nIt's a test mail!\n");
212     smtpClient.connect();
213     smtpClient.sendMail();

```

SMTPClient.java

```
214     smtpClient.disconnect();
215     System.out.println("OK");
216 }
217
218 }
219
```