

/*POP Client :gives the server name,username and password,retrieve the mails and allow manipulation of mailbox using POP commands*/

```
import java.io.*;
import java.net.*;
import java.awt.*;
import javax.swing.*;
import java.util.*;
import java.awt.event.*;
```

```
class PopServer extends JFrame implements ActionListener
{
```

```
    JPanel jp;
    JLabel lpadd,lpnum,lpass,lretr,luser;
    JTextField padd,pnum,user,retr,del;
    JPasswordField pass;
    JTextArea receive;
    JScrollPane scrIp;
    JButton login,list,quit,retrv,dele;
    Socket s;
    PrintWriter pw;
    BufferedReader br;
    int nmsgs;
```

```
    PopServer()
```

```
    {
        jp=new JPanel();
        jp.setLayout(null);

        lpadd=new JLabel("port address");
        lpadd.setBounds(20,20,100,20);
        jp.add(lpadd);

        padd=new JTextField();
        padd.setBounds(110,20,120,20);
        jp.add(padd);

        lpnum=new JLabel("port number");
        lpnum.setBounds(20,40,100,20);
        jp.add(lpnum);

        pnum=new JTextField();
        pnum.setBounds(110,40,120,20);
        jp.add(pnum);

        login=new JButton("login");
        login.setBounds(20,100,210,20);
        jp.add(login);

        login.addActionListener(this);
        luser=new JLabel("user name");
        luser.setBounds(20,60,100,20);
        jp.add(luser);

        user=new JTextField();
        user.setBounds(110,60,120,20);
        jp.add(user);
```



```

lpass=new JLabel("password");
lpass.setBounds(20,80,100,20);
jp.add(lpass);

pass=new JPasswordField();
pass.setBounds(110,80,120,20);
jp.add(pass);

list=new JButton("list");
list.setBounds(20,120,210,20);
list.addActionListener(this);
jp.add(list);

retrv=new JButton("retrieve");
retrv.setBounds(130,145,130,20);
jp.add(retrv);

retrv.addActionListener(this);
dele=new JButton("delete");
dele.setBounds(130,165,130,20);
jp.add(dele);

dele.addActionListener(this);
lretr=new JLabel("enter the meg numebr");
lretr.setBounds(20,145,120,20);
jp.add(lretr);

retr=new JTextField();
retr.setBounds(100,145,30,20);
retr.addActionListener(this);
jp.add(retr);

del=new JTextField();
del.setBounds(100,165,30,20);
del.addActionListener(this);
jp.add(del);

receive=new JTextArea();
receive.setEditable(false);

scrlp=new JScrollPane(receive);
jp.add(scrlp);
scrlp.setBounds(20,200,300,200);

quit=new JButton("quit");
quit.setBounds(130,410,80,30);
jp.add(quit);
quit.addActionListener(this);

setTitle("Post Office Protocol");
setSize(350,470);
show();
this.getContentPane().add(jp);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{

```



```

        if(ae.getSource()==login)
        {
            try
            {
                s=new Socket(padd.getText(),Integer.parseInt(pnum.getText()));
br=new BufferedReader(new InputStreamReader(s.getInputStream()));
                pw=new PrintWriter(new OutputStreamWriter(s.getOutputStream()),true);
                receive.setText(br.readLine()+"\n");
                pw.println("user"+user.getText());
                receive.append(br.readLine()+"\n");
                pw.println("pass"+pass.getText());
                receive.append(br.readLine()+"\n");
            }
            catch(IOException e)
            {
                JOptionPane.showMessageDialog(new JPanel(),"connection can not
be established");
            }
        }

        if(ae.getSource()==list)
        {
            StringTokenizer st;
            String str;
            pw.println("list");
            try
            {
                str=br.readLine();
                st=new StringTokenizer(str);
                st.nextToken();
                str=st.nextToken();
                nmsgs=Integer.parseInt(str)+1;
                for(int i=0;i<nmsgs;i++)
                receive.append(br.readLine()+"\n");
                receive.append("no of messages : " + str + "\n");
            }
            catch(Exception e)
            {
                JOptionPane.showMessageDialog(new JPanel(),e);
            }
        }

        if(ae.getSource()==retrv)
        {
            String k=retr.getText().trim();
            pw.println("RETR \n"+k);
            int l=Integer.parseInt(k);
            try
            {
                String st=br.readLine();
                if(l<nmsgs&&l>0)
                {
                    while(!st.equals("."))
                    {
                        receive.append(st+"\n");
                        st=br.readLine();
                    }
                }
            }
        }
    }
}

```



```

        }
        else
        {
            receive.append(st+"\n");
        }
        retr.setText("");
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(new JPanel(),e);
    }
}
if(ae.getSource()==quit)
    System.exit(0);
if(ae.getSource()==dele)
{
    try
    {
        pw.println("dele"+del.getText().trim());
        nmsgs--;
        del.setText("");
        receive.append(br.readLine()+"\n");
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(new JPanel(),e);
    }
}
}

public static void main(String args[])
{
    PopServer p=new PopServer();
}
} //end of PopServer class

```

