**Email User Agent项目实验报告**

1. **项目需求**

（一）web服务器应该响应来自用户代理正确的http请求。

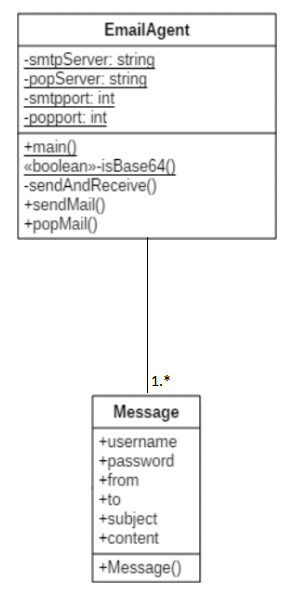
（二）web服务器应该能够同时服务至少2个http用户代理。

（三）用户可以设置此web服务器的工作端口。

1. **开发环境与工具**

本项目使用java及jdk1.8进行开发，编码及测试主要使用eclipse作为工具。

1. **详细设计**
2. UML类图



1. 详细编码

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.InetAddress;

import java.net.Socket;

import java.util.Scanner;

import javax.net.ssl.SSLSocketFactory;

import sun.misc.BASE64Encoder;

import sun.misc.BASE64Decoder;

public class EmailAgent {

private static String smtpServer = "smtp.163.com";

private static String popServer = "pop.163.com";

private static int smtpport = 465;

private static int popport = 995;

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

String account = null;

String password = null;

while(true){

System.out.print("请输入您的邮箱账号(只支持qq、163邮箱、新浪邮箱)：");

account = input.nextLine();

if(account.indexOf("@qq.com") != -1){

smtpServer = "smtp.qq.com";

popServer = "pop.qq.com";

break;

}else if(account.indexOf("@163.com") != -1){

smtpServer = "smtp.163.com";

popServer = "pop.163.com";

break;

}else if(account.indexOf("@sina.cn") != -1){

smtpServer = "smtp.sina.cn";

popServer = "pop.sina.cn";

break;

}else{

System.out.println("未识别此邮箱，请重新输入");

}

}

while(true){

System.out.print("请输入密码：");

password = input.nextLine();

Socket socket = null;

try{

socket = SSLSocketFactory.getDefault().createSocket(smtpServer,smtpport);

socket.setSoTimeout(10000);

BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter pw = new PrintWriter(socket.getOutputStream(),true);

String localhost = InetAddress.getLocalHost().getHostName();

String base64account = new BASE64Encoder().encode(account.getBytes());

String base64password = new BASE64Encoder().encode(password.getBytes());

System.out.println("Server>"+br.readLine());

pw.println("HELO "+localhost);

String temp = br.readLine();

if(temp.indexOf("250") == -1){

System.out.println("无法连接服务器");

}

pw.println("AUTH LOGIN");

temp = br.readLine();

pw.println(base64account);

temp = br.readLine();

pw.println(base64password);

temp = br.readLine();

System.out.println("Server>"+temp);

pw.println("QUIT");

if(temp.indexOf("235") != -1){

System.out.println("密码验证成功");

socket.close();

break;

}else{

System.out.println("密码验证失败");

}

}catch (IOException e){

e.printStackTrace();

}finally{

if(socket !=null){

try {

socket.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

while(true){

String option = null;

System.out.println("请选择：quit:退出|send:发邮件|receive:收邮件");

option = input.nextLine();

if(option.equals("quit")){

if(input != null){

input.close();

}

break;

}

switch(option){

case "send":

System.out.print("请输入收件人地址：");

String address = input.nextLine();

System.out.print("请输入邮件主题：");

String subject = input.nextLine();

System.out.print("请输入邮件内容：");

String content = input.nextLine();

Message msg = new Message(account, password, account, address, subject, content);

new EmailAgent().sendMail(msg);

break;

case "receive":

new EmailAgent().popMail(account, password, input);

break;

default:

System.out.println("未识别序号");

break;

}

}

// Message msg = new Message("1324908543@163.com", "1324908543@qq.com", "测试1", "TEST2");

// new EmailAgent().sendMail(msg);

// new EmailAgent().popMail();

}

private static boolean isBase64(String str) {

if (str == null || str.length() == 0) {

return false;

} else {

if (str.length() % 4 != 0) {

return false;

}

char[] strChars = str.toCharArray();

for (char c:strChars) {

if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0' && c <= '9')

|| c == '+' || c == '/' || c == '=') {

continue;

} else {

return false;

}

}

return true;

}

}

private void sendAndReceive(String str,BufferedReader br,PrintWriter pw) throws IOException {

if(str != null){

System.out.println("Client>"+str);

pw.println(str);

}

String response;

if((response = br.readLine())!=null){

System.out.println("Server>"+response);

}

}

public void sendMail(Message msg){

Socket socket = null;

try {

socket = SSLSocketFactory.getDefault().createSocket(smtpServer,smtpport);

socket.setSoTimeout(10000);

BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter pw = new PrintWriter(socket.getOutputStream(),true);

String localhost = InetAddress.getLocalHost().getHostName();

// String username = "1324908543@qq.com";

// String password = "cwbohgxzjwksbdhf";

// String username = "1324908543@163.com";

// String password = "cyhxdnflxsqqlxs1";

// username = new BASE64Encoder().encode(username.getBytes());

// password = new BASE64Encoder().encode(password.getBytes());

String base64account = new BASE64Encoder().encode(msg.username.getBytes());

String base64password = new BASE64Encoder().encode(msg.password.getBytes());

System.out.println("Server>"+br.readLine());

sendAndReceive("HELO "+localhost, br, pw);

sendAndReceive("AUTH LOGIN", br, pw);

sendAndReceive(base64account, br, pw);

sendAndReceive(base64password, br, pw);

sendAndReceive("MAIL FROM:<"+msg.from+">", br, pw);

sendAndReceive("RCPT TO:<"+msg.to+">", br, pw);

pw.println("DATA");

System.out.println("Client>"+ "DATA");

pw.println("From:"+msg.from);

System.out.println("Client>"+"From:"+msg.from);

pw.println("To:"+msg.to);

System.out.println("Client>"+"To:"+msg.to);

pw.println("Subject:"+ msg.subject+"\r\n");

System.out.println("Client>"+"Subject:"+msg.subject);

pw.println(msg.content);

System.out.println("Client>"+ "Content:" + msg.content);

sendAndReceive(".", br, pw);

sendAndReceive("QUIT", br, pw);

} catch (IOException e) {

e.printStackTrace();

} finally{

if(socket!=null){

try {

socket.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

public void popMail(String account, String password, Scanner input){

Socket socket = null;

try {

socket = SSLSocketFactory.getDefault().createSocket(popServer, popport);

BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter pw = new PrintWriter(socket.getOutputStream(),true);

System.out.println("Server>"+br.readLine());

String username = "-@qq.com";

String password = "cwbohgxzjwksbdhf";

String username = "-@qq.com";

String password = "cwbohgxzjwksbdhf"; //qq授权码cwbohgxzjwksbdhf

sendAndReceive("user "+account, br,pw); //用户名

sendAndReceive("PASS "+password, br,pw); //用户名

pw.println("stat");

String temp[] = br.readLine().split(" ");

int count = Integer.parseInt(temp[1]);//得到信箱中共有多少封邮件

System.out.println("您一共有" + count + "封邮件字收件箱");

while(true){

System.out.println("请输入：quit:退出|ckeck:查看邮件内容");

String next = input.nextLine();

if(next.equals("quit")){

break;

}

switch(next){

case "quit":

break;

case "check":

System.out.print("您希望打开第几封？(最新的邮件排在最后)>");

int open = Integer.parseInt(input.nextLine());

System.out.println("请输入字符集，如GBK, UTF-8：");

String charset = input.nextLine();

pw.println("retr " + open);

System.out.println("第" + open + "封邮件的内容:");

while (true) {

String reply = br.readLine();

if(isBase64(reply)){

byte[] bt = new BASE64Decoder().decodeBuffer(reply);

reply = new String(bt,charset);

}

System.out.println(reply);

if (reply.toLowerCase().equals(".")) {

break;

}

}

break;

default:

System.out.println("未识别操作");

break;

}

}

} catch (IOException e) {

System.out.println(e.toString());

} finally {

try {

if (socket != null ) {

socket.close();

}

} catch (IOException e) {}

}

}

}

class Message{

String username;

String password;

String from;

String to;

String subject;

String content;

public Message(String username, String password, String from,String to,String subject,String content){

this.username = username;

this.password = password;

this.from = from;

this.to = to;

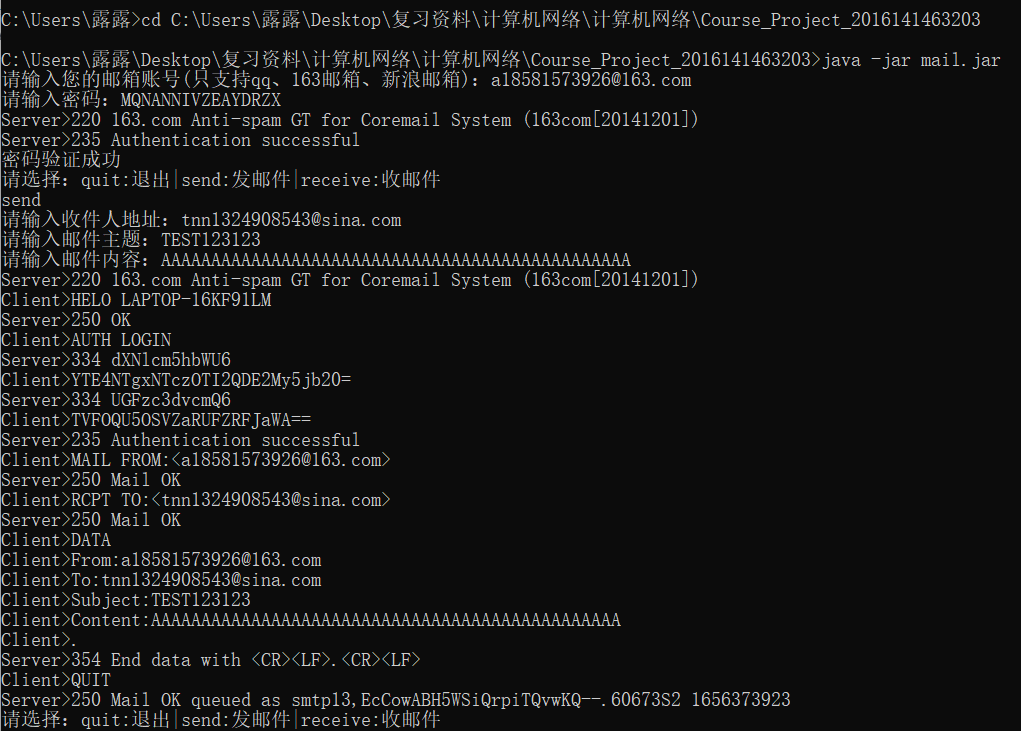
this.subject = subject;

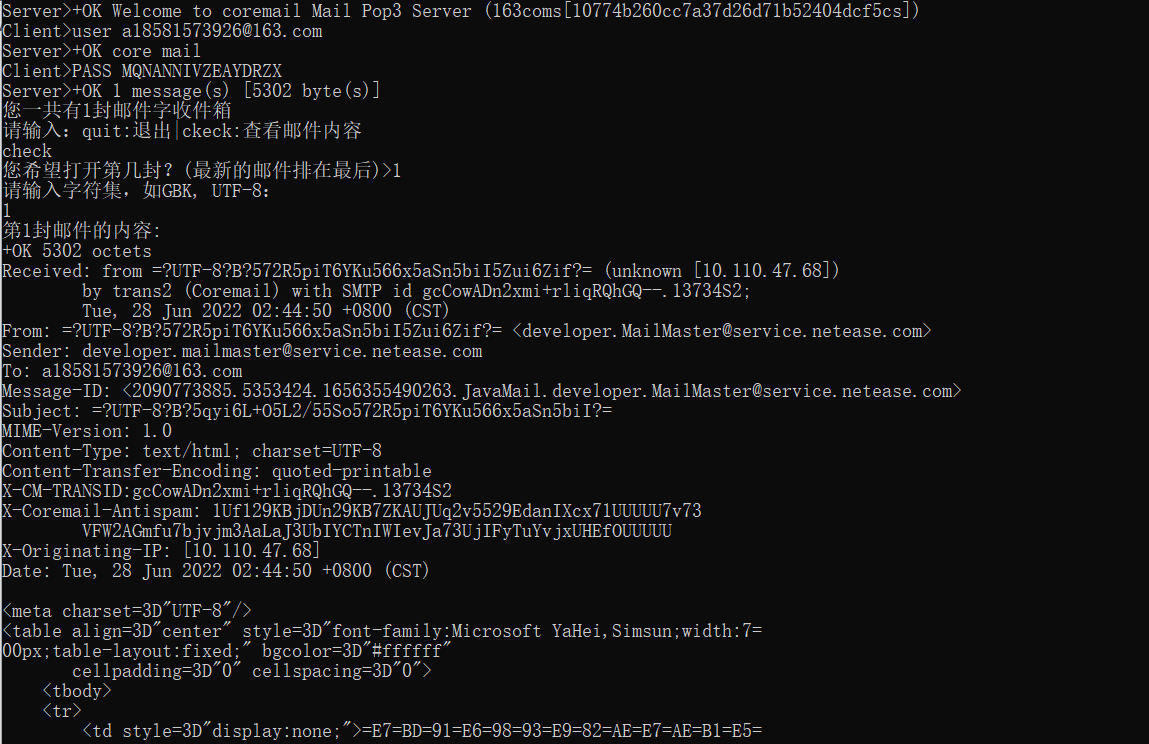
this.content = content;

}

}

运行截图：





**（三）程序设计**

1. 主类EmailAgent

在主类EmailAgent中，设置了4个静态成员变量，分别为String smtpServer；String popServer；int smtpport；int popport。它们分别用于存储smtp服务器、pop服务器、smtp服务器端口号、pop服务器端口号，默认值分别为”smtp.163.com”, “pop.163.com”, 465, 995。

由于在创建Socket时，使用了SSL加密，因此端口号设置为465和995，否则smtp端口号为25，pop端口号为110。

主函数 main()：内部有三个成员变量Scanner input; String account; String password;作用分别是读取用户输入，存储用户账号，密码。函数有三个while循环，第一个用于判断用户输入的账号类别，是qq邮箱、163邮箱还是新浪邮箱；第二个判断所选服务器是否可连接，密码是否正确；第三个循环用于根据用户选择，进行邮件的收发和退出。

函数boolean isBase64(String str)：用于判断读入的文本是否是Base64编码，因为经Base64编码后文本的每个字符，可能的值都在A-Z、a-z、0-9、+、/内，所以本函数就进行了简单的判断，若文本字符均在上述字符范围内，就可以认定为经过了Base64编码。

函数sendAndReceive(String str,BufferedReader br,PrintWriter pw)：用于向服务器发送一条指令，并接受服务器响应，str为输入的指令。

void sendMail(Message msg)：用于发送邮件。参数为一个Message对象，获取账户、密码、源地址、目的地址、邮件主题和邮件内容。

void popMail(String account, String password, Scanner input)：用于邮件的接收。参数为String account, String password, Scanner input，获取账号、密码及输入。

1. Message

其中有6个成员变量：String username;String password;String from;String to; String subject; String content;分别用于存储账号、密码、源地址、目的地址、主题、邮件。

构造函数Message(String username, String password, String from,String to,String subject,String content)

**（四）项目总结**

本项目已基本实现预计功能，可通过运行jar包实现登入账户，实现用户名与密码的输入、收发邮件等功能。

本项目基本原理为：通过SSL加密，连接163、qq、新浪的smtp、pop服务器；通过smtp服务器发送纯文本邮件，给163、qq、新浪邮箱；从pop服务器接收文本邮件，并通过使用者指定的字符集进行解析。

目前本项目存在的问题有：登录邮箱时需要使用授权码；在使用部分邮箱（如163邮箱）发送邮件时可能被判定为垃圾邮件；在收取邮件时，用户需要确定所收取邮件的字符集，否则会出现乱码。