# 第7讲 邮件发送程序设计

**教学与实践目的：**学会网络邮件发送的程序设计技术。

网络程序设计的精髓是什么？

就是客户端和服务器的对话和响应契约(协议)。 简单邮件传输协议SMTP可以很好地诠释这一点。

邮件传输协议包括SMTP（简单邮件传输协议，RFC821）及其扩充协议MIME；

邮件接收协议包括POP3和功能更强大的IMAP协议。

服务邮件发送的服务器其端口为25（开启ssl可能使用465或587端口），服务邮件接收的服务器端口为110。

邮箱设置一定要开启smtp/pop3服务，目前大部分邮箱开启服务时，需要设置第三方客户端使用的授权码，该授权码就是代替密码使用，目的是防止密码泄露， qq邮箱、163邮箱等均改成了这种使用方式，例如图7.1所示。

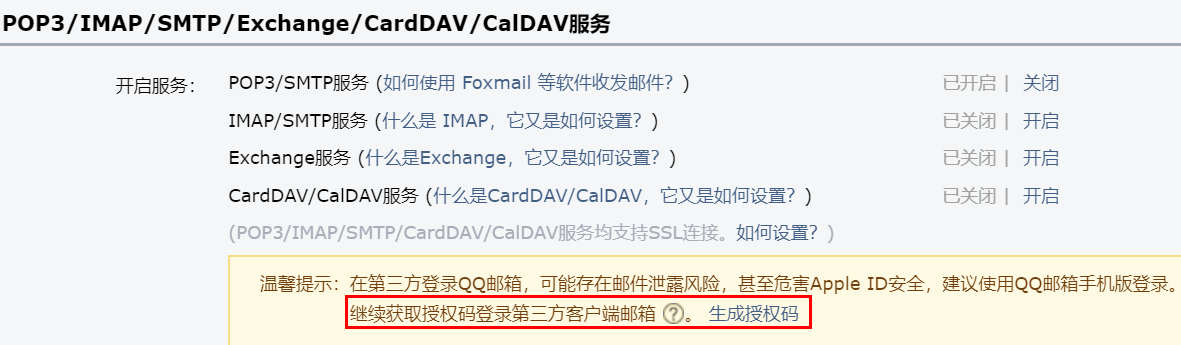


图7.1 QQ邮箱的授权码

## 一、 程序设计第一步：准备一段BASE64编码程序

新建一个Java包，建议包名：chapter07，并新建BASE64.java程序。

SMTP过程中，要成功地发送和接收邮件，用户名和密码需要BASE64编码后才能有效传输，这个类就是完成这个需求：

public class BASE64 {  
 public static void main(String[] args) {  
 String userName="你的完整邮箱名";  
 String authCode = "你的邮箱授权码";  
 *//显示邮箱名的base64编码结果* System.*out*.println(*encode*(userName));  
 *//显示授权码的base64编码结果* System.*out*.println(*encode*(authCode));  
 }  
  
 public static String encode(String str) {  
 return new sun.misc.BASE64Encoder().encode(str.getBytes());  
 }  
}

**注意：**

（1）若用户密码验证过程中出现失败或提示需要开启SMTP终端服务，则通过浏览器的 web方式进入你的邮箱，更改邮箱的属性设置-->帐户,勾选或开启支持SMTP客户端软件；

（2）对于QQ邮箱还需要先设置独立密码，然后才能开启支持SMTP客户端软件（这个工作要提前一周做）。

## 二、telnet手动体验SMTP协议过程1

操作系统默认没有安装telnet，可以通过控制面板=》程序和功能=》启用或关闭windows功能中选择telnet客户端进行安装。安装后，打开操作系统的命令行窗口，键入：telnet smtp.qq.com 25（完成三次握手连接），然后完成如下的对话过程，其中加粗加亮字体是smtp协议支持的命令，必须输入一致，其余内容根据具体情况填写，双斜杠内容是注释，不用输入：

**HELO**  hostname //hostname可以是IP或其他随意别名

**AUTH LOGIN**

//在输入auth login回车后，先粘贴base64程序编码的完整邮箱名并回车；再粘贴base64编码的授权码并回车

**MAIL FROM:<your mail address>** //在这里填写自己的邮箱地址，用于发送邮件（注意冒号后面别有空格，建议使用注册时的邮箱地址）

**RCPT TO:<your mail address>** //接收方的邮箱，在这里暂时填写和上面一样的邮箱地址，即自己发送邮件给自己，验证是否成功

**DATA** //在输入data回车后，接下来开始发送邮件头相关内容

**Subject:** this is simple mail // 邮件的标题

//在这里再多发送一行空行，来分隔邮件内容，下面就是邮件正文内容

Hello!

很好！

//控制台中中文会变成?的乱码，但不影响接受者接收到中文内容

**.** //在邮件正文发送完毕后，单独用一行输入一个小圆点，作为结束标志，然后回车

**QUIT** //结束通信（含4次握手断开）

知识点：以上基于命令行窗口的邮件发送方法，可以让你不需要其他软件的帮助就可以完成简单邮件的发送与接收。

## 三、程序设计第二步：体验SMTP协议过程2

现在用自己的程序代替telnet，来验证smtp发送给邮件的过程：

（1）将第3讲原有的TCPClient.java程序复制重构到chapter07程序包下；

（2）将第3讲TCPClientThreadFX.java对话窗口程序，复制到chapter07程序包下，并重构命名为TCPMailClientFX.java，并添加针对用户名和密码的Base64编码功能；

（3）运行TCPMailClientFX.java程序，ip文本框填写smtp.qq.com，端口文本框填写25，连接后按照如下的对话顺序通过send按钮发送命令，注意，和第一次体验有所不同，增加了发件人信息和收件人信息：

**HELO**  hostname //hostname可以是IP或其他随意别名

**AUTH LOGIN**

//发送auth login后，先发送Base64编码的邮箱名，然后再发送Base64编码的授权码

**MAIL FROM:<your mail address>** //在这里填写《互联网程序设计》实验注册的邮箱地址，用于发送邮件（注意冒号后面别有空格）

**RCPT TO:<other mail address>** //接收方的邮箱，在这里用自己另外的邮箱或同学的邮箱地址测试。

**DATA** //在输入data回车后，接下来开始设置邮件头，邮件头除了邮件标题，还包括发件人地址和收件人地址

**From:\*\*\*\*\*\*@qq.com** // 发件人的邮箱地址

**Subject:** **给同学的一封测试邮件** // 邮件的标题

**To:\*\*\*\*@qq.com** //会显示在邮件接收者栏上

// 在这里再多发送一行空行，来分隔邮件内容，下面是邮件正文内容

I am \*\*\*同学 //邮件正文内容

**.** //在邮件正文发送完毕后，单独发送一个小圆点，作为结束标志

**QUIT** //发送quit结束通信（含4次握手断开）

用自己的图形界面进行smtp交互发送邮件，是否比telnet方便？

**注意：**

判断邮件是否成功发送，可查看你“已发送”箱中该邮件的发送状态或是否有发送失败的提示或收到自动回复邮件（仅供参考）。

## 四、窗口程序实现自动发邮件

（1）设置如图7.2所示的邮件发送窗口（供参考），并命名为TCPMailClientFX2.java，将手动操作的SMTP过程封装到程序中，实现邮件自动发送，注意不要短时间频繁发送，会被邮件服务器拒绝smtp服务。

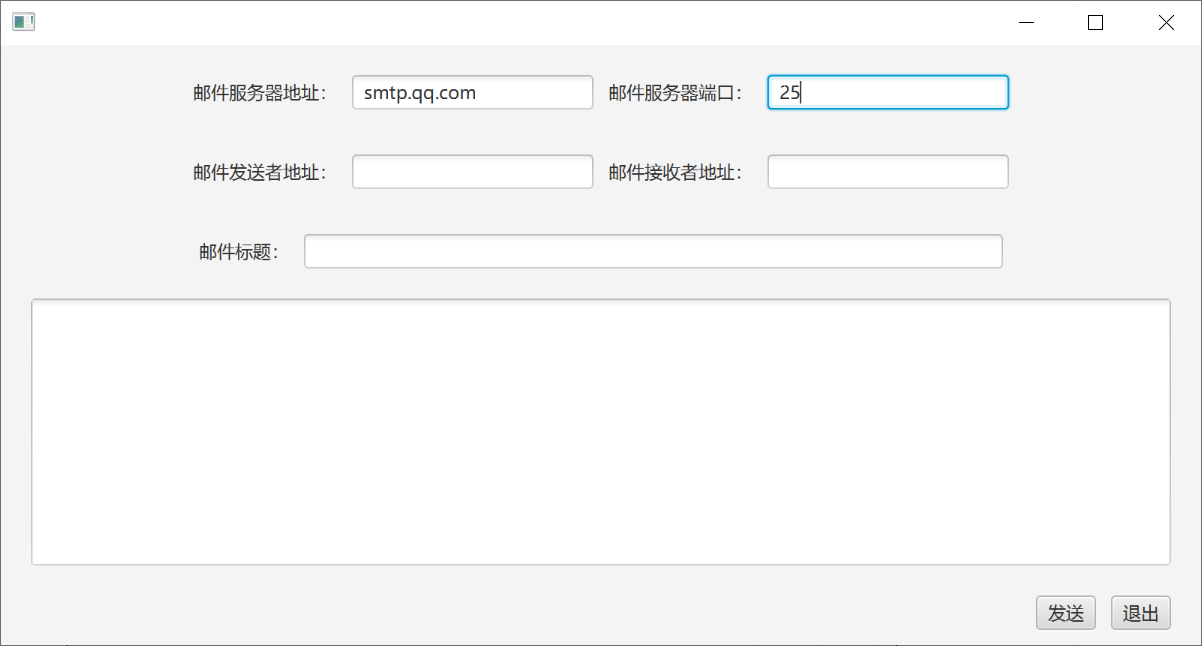


图7.2 邮件发送窗口

（2）将TCPClient.java复制重构，并命名为TCPMailClient.java,并修改其中的发送方法如下：

public void send(String msg) {  
 *//输出字符流，由Socket调用系统底层函数，经网卡发送字节流* pw.println(msg);  
 try {  
 *//进行邮件交互、发送smtp指令之间应该暂停一段时间* Thread.*sleep*(200);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
}

注意：邮件命令需要一行一行地发送，使用程序自动发送过程中需要设置一定的睡眠等待时间。

（3）“发送按钮”的部分代码如下：

String smtpAddr = tfSmtpAddr.getText().trim();  
String smtpPort = tfSmtpPort.getText().trim();  
try {  
 tcpMailClient = new TCPMailClient(smtpAddr, smtpPort);  
 tcpMailClient.send("HELO myfriend");  
  
 tcpMailClient.send("AUTH LOGIN");  
  
 String userName = "你的完整邮箱地址";  
 String authCode = "申请的授权码";  
  
 String msg = BASE64Encoder.*encode*(userName);  
 tcpMailClient.send(msg);  
  
 msg = BASE64Encoder.*encode*(authCode);  
 tcpMailClient.send(msg);  
  
 msg = "MAIL FROM:<" + tfSenderAddr.getText().trim() + ">";  
 tcpMailClient.send(msg);  
  
 msg = "RCPT TO:<" + tfRecieverAddr.getText().trim() + ">";  
 tcpMailClient.send(msg);  
  
 msg = "DATA";  
 tcpMailClient.send(msg);  
  
 msg = "FROM:" + tfSenderAddr.getText().trim();  
 tcpMailClient.send(msg);

……

……

## 五、改善TCPMailClientFX2(课堂计分)

（1）TCPMailClientFX2看不到服务器反馈信息，不利于了解发送状态。例如自动发送邮件的程序，如果短时间发送次数过多，有时候邮件服务器会出现502错误，造成发送失败，所以有必要添加服务器反馈信息的显示。修改程序，添加一个文本区域，用于显示邮件服务器的反馈信息，并尽量解决同时发送和显示服务器信息造成的卡顿。参考界面如图7.3所示。

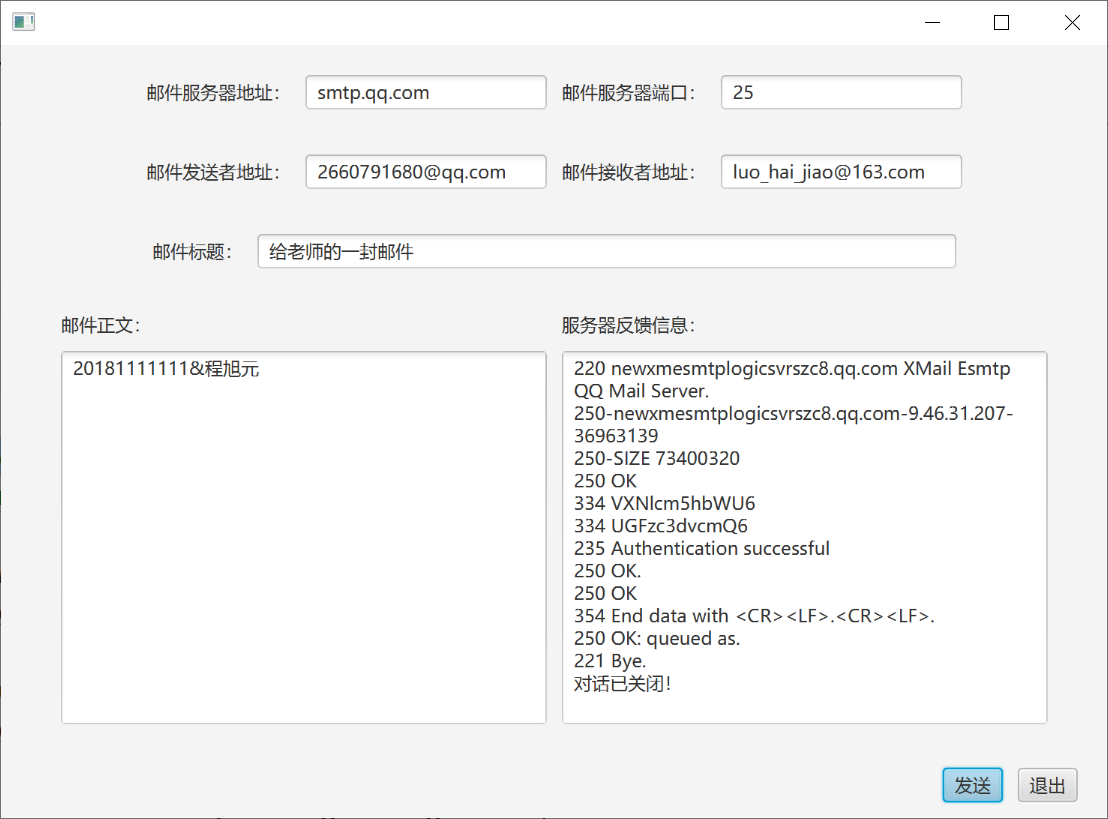


图7.3 带服务器反馈信息的界面

（2）邮件发送者地址应该是《互联网程序设计》实验注册的邮件地址，邮件接收者地址为：luo\_hai\_jiao@163.com，发送内容必须为：学号&姓名，例如20181111111&程旭元。用该程序发送这个指定邮件后，要知是否发生成功，除了观察服务器反馈信息外，还可查看自己的发送邮箱是否收到自动回复邮件。如果有收到回复信息，可再次查询服务器是否获得课程分。

特别要注意：邮件发送者地址、邮件接收者地址和邮件发送的内容必须按规定填写，没得到分数往往是上面的失误造成的。

## 六、扩展练习一 验证POP3(110)电子邮件接收协议

可以使用telnet 或 TCPMailClientFX.java来验证，这里以telnet为例：

**TELNET** pop.qq.com 110

**USER** 完整邮箱名 //不需要base64编码

**PASS** 授权码 //不需要base64编码

**STAT** 或**LIST** //stat统计当前邮箱数量和占用空间，list列出所有邮件信息，如果邮件特别多，list会比较耗时

**RETR** 2 //显示第2个邮件，注意，显示的都是编码的内容，正常无法阅读

**QUIT** //退出

## 七、扩展练习二 JavaMail API 程序设计

为了简化邮件客户程序的开发，Oracle公司制定了JavaMail API，它封装了按照各种邮件通信协议，如IMAP、POP3和SMTP等与邮件服务器通信的细节，为Java应用程序提供了收发电子邮件的公共接口，用于开发邮件收发客户端，有兴趣的同学可进一步学习。

（1）下载JavaMail API类库文件javax.mail.jar并加入到项目：

对于加入外部jar，在项目开发中经常使用maven来进行配置，但对于我们的练习项目，完全可以手动下载jar，添加到项目中。该类库文件下载地址为：<https://github.com/javaee/javamail/releases/download/JAVAMAIL-1_6_2/javax.mail.jar>。对于idea，可以在项目工程下新建lib文件夹，将需要的外部jar包都拷贝到这个文件夹，如图7.4所示，

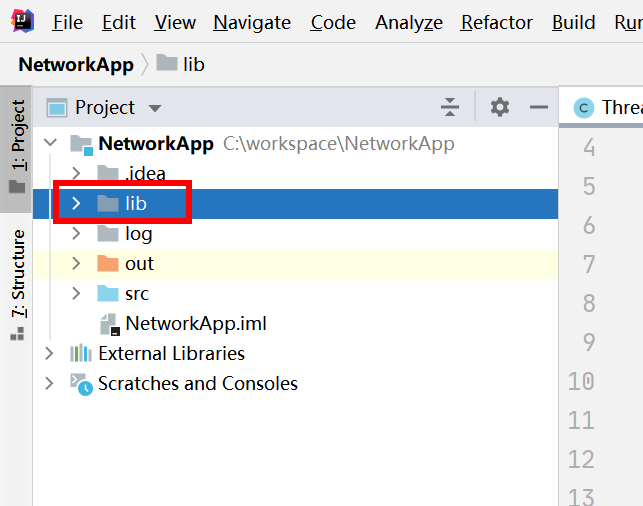


图7.4 新建lib文件夹

这一步结束还没有将lib下的jar导入到项目，需要进一步设置，推荐用图7.5所示方式：

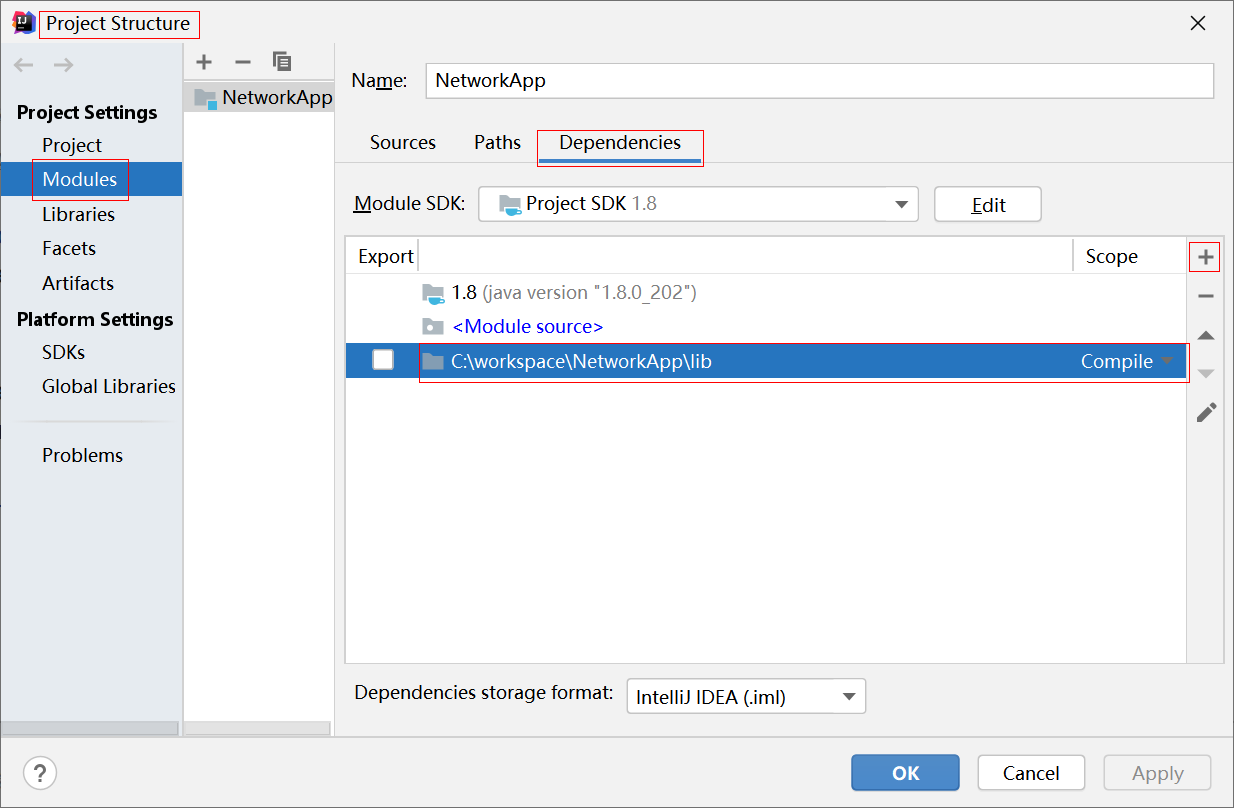


图7.5 导入jar包

（2）阅读理解程序文件JavaMailAPIClient.java，将对应变量修改为自己的邮箱信息；

（3）运行及调试，能成功发送和接收邮件。

JavaMailAPIClient.java源程序：

import javax.mail.\*;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;  
import java.util.Properties;  
  
public class JavaMailAPIClient {  
 */\*  
 Session类表示邮件会话，是JavaMail API的最高层入口类。  
 Session对象从java.util.Properties对象中获取配置信息，  
 如邮件发送服务器的主机名或IP地址、接收邮件的协议、发送邮件的协议、  
 用户名、口令及在整个应用程序中共享的其他信息。  
 \* \*/* protected Session session;  
 */\*  
 Store类表示接收邮件服务器上的注册账号的存储空间，  
 通过Store类的getFolder()方法，可以访问用户的特定邮件夹。  
 \*/* protected Store store;  
 */\*  
 Transport类根据指定的邮件发送协议（通常是 SMTP），  
 通过指定的邮件发送服务器来发送邮件。  
 Transport类是抽象类，它的静态方法send(Message)负责发送邮件  
 \* \*/* protected Transport transport;  
  
 *//定义接收邮件服务器* private String receiveHost = "pop.qq.com";  
 *//定义发送邮件服务器* private String sendHost = "smtp.qq.com";  
 *//接收邮件协议* private String receiveProtocol = "pop3";  
  
  
 public void init() throws Exception {  
 *//设置邮箱属性* Properties props = new Properties();  
 props.put("mail.transport.protocol", "smtp");  
 props.put("mail.store.protocol", receiveProtocol);  
 props.put("mail.imap.class", "com.sun.mail.imap.IMAPStore");  
 props.put("mail.smtp.class", "com.sun.mail.smtp.SMTPTransport");  
 props.put("mail.smtp.host", sendHost);*//设置发送邮件服务器  
  
 //创建session对象* session = Session.*getDefaultInstance*(props);  
  
 *//创建Store对象* store = session.getStore(receiveProtocol);  
  
 *//连接到接收邮件服务器* store.connect(receiveHost, "接收邮箱的完整地址", "该邮箱的授权码");  
 }  
  
 public void close() throws Exception {  
 store.close();  
 }  
  
 */\*\*  
 \* 发送邮件  
 \* @param fromAddr 发送邮件的邮箱地址  
 \* @param toAddr 接收邮件的邮箱地址  
 \*/* public void sendMail(String fromAddr, String toAddr)  
 throws MessagingException {  
 *//发送邮件时，我们需要构造一个Message对象，  
 // 然后调用Transport.send(Message)即可完成发送* MimeMessage message = new MimeMessage(session);  
 *// 设置发送方地址* message.setFrom(new InternetAddress(fromAddr));  
 *// 设置接收方地址* message.setRecipient(Message.RecipientType.*TO*, new InternetAddress(toAddr));  
 *// 设置邮件主题* message.setSubject("你好", "UTF-8");  
 *// 设置邮件正文* message.setText("开心的一天！", "UTF-8");  
 *//发送* Transport.*send*(message,fromAddr,"该发送邮箱的授权码");  
 }  
  
 */\*\*  
 \* 接收邮件  
 \* @throws Exception  
 \*/* public void receiveMail() throws Exception {  
 *//inbox邮件夹是邮件账号的保留邮件夹(IMAP不允许用户删除该邮件夹)，  
 // 邮件服务器把所有接收到的新邮件都存在该邮件夹中* browseMessagesFromFolder("inbox");  
 }  
  
 public void browseMessagesFromFolder(String folderName) throws Exception {  
 Folder folder = store.getFolder(folderName);  
 if (folder == null) {  
 throw new Exception(folderName + "邮件夹不存在");  
 }  
 browseMessagesFromFolder(folder);  
 }  
  
 public void browseMessagesFromFolder(Folder folder) throws Exception {  
 *//Folder类代表邮件夹，邮件都放在邮件夹中* folder.open(Folder.*READ\_ONLY*);  
 System.*out*.println("你的邮箱里有： " + folder.getMessageCount() + "封邮件");  
 System.*out*.println("你的邮箱里有： " + folder.getUnreadMessageCount() + "封未读邮件");  
  
 *//读邮件,Message类代表电子邮件。Message类提供了读取和设置邮件内容的方法* for (int i = 1; i <= folder.getMessages().length; i++) {  
 System.*out*.println("-------第" + i + "封邮件--------");  
 System.*out*.print("其发送者是:");  
  
 *//Address类代表邮件地址，和Message 类一样，Address类也是个抽象类。  
 // 常用的具体子类为javax.mail.internet.InternetAddress类* InternetAddress[] addr = (InternetAddress[]) folder.getMessage(i).getFrom();  
  
 *//发送者地址* System.*out*.println(addr[0].getAddress());  
 System.*out*.print("发送的标题为：");  
 String subject = folder.getMessage(i).getSubject();  
 System.*out*.println(subject);  
 System.*out*.print("发送日期为：");  
  
 System.*out*.println(" 邮件内容为： ");  
 System.*out*.println(folder.getMessage(i).getContent().toString());  
 }  
  
 folder.close(false);*//关闭邮件夹，但是不要删除邮件夹中标记为"deleted"的邮件* }  
  
  
 public static void main(String[] args) throws Exception{  
 JavaMailAPIClient mailClient = new JavaMailAPIClient();  
 mailClient.init();  
 mailClient.sendMail("your-test@qq.com","your-test@qq.com");  
*// mailClient.receiveMail();* mailClient.close();  
 }  
}