

NGÔN NGỮ LẬP TRÌNH JAVA

Nội dung

Cơ bản về ngôn ngữ lập trình Java

Lập trình hướng
đối tượng

Biến, từ khoá,
kiểu dữ liệu

Biểu thức, các
cấu trúc điều
khiển

Dữ liệu kiểu
mảng

Các khía cạnh nâng cao của lập trình hướng đối tượng

Thiết kế lớp

Thiết kế lớp
nâng cao

Xử lý ngoại lệ

Generics

Java Collection
Framework

Multithread&
Concurrency

Database
Programming

Network
Programming

Send mail

SEND MAIL

Nội dung

1. Warriors of the NET
2. SMTP
3. Send & receive email from a gmail account

1. Warriors of the net

Warriors of the Net (1999)

[SEE RANK](#)

13 min - [Short](#)



Your rating: ★★★★★★★★★★ -/10

Ratings: **7,6**/10 from [52 users](#)

Reviews: [2 user](#)

For the first time in history people and machinery are working together, realizing a dream. A uniting force that knows no geographical boundaries. Without regard to race, creed or color. A ... [See full summary](#) »

Writer: [Tomas Stephanson](#)

Stars: [Monte Reid](#) | [See full cast and crew](#) »

+ Watchlist



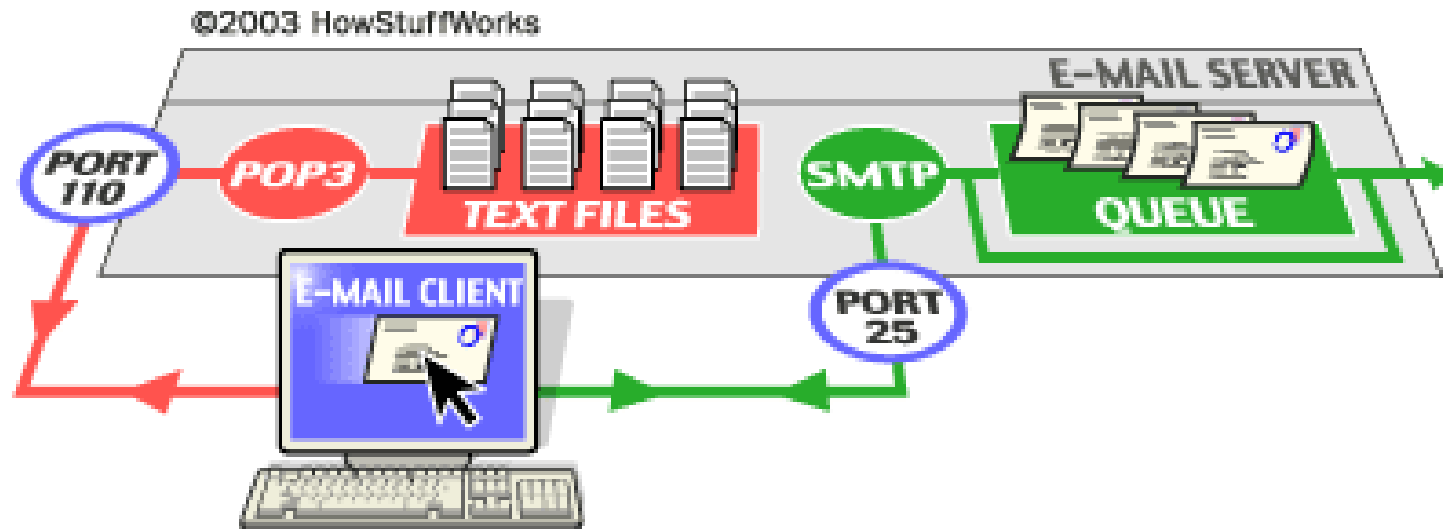
Share...

[Own the rights? Add a poster](#) »

[More at IMDbPro](#) »

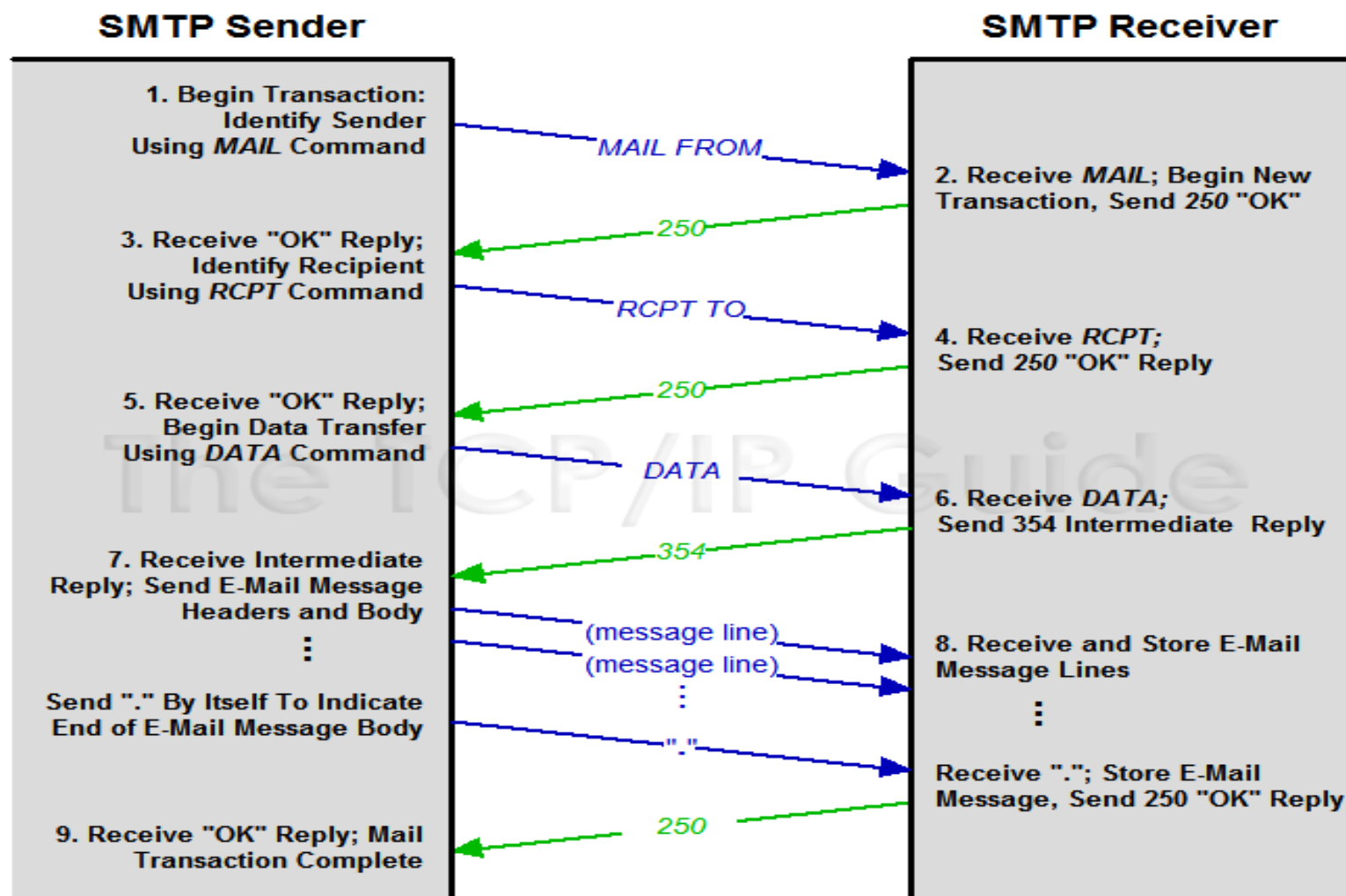
2. Simple Mail Transfer Protocol (1/)

- ❑ What is SMTP?
 - ❑ Simple Mail Transfer Protocol
 - ❑ A TCP/IP protocol used in sending and receiving email



2. Simple Mail Transfer Protocol (2/)

SMTP MAIL processing Model



2. Simple Mail Transfer Protocol (3/)

□ Transaction example

- **MAIL FROM:**<joe@someplace.org>
250 <joe@someplace.org>... Sender ok
RCPT TO:<jane@somewhereelse.com>
250 <jane@somewhereelse.com>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
From: Joe Sender <joe@someplace.org>
To: Jane Receiver <jane@somewhereelse.com>
Date: Sun, 1 Jun 2003 14:17:31 —0800
Subject: Lunch tomorrow

Hey Jane,

It's my turn for lunch tomorrow. I was thinking we could
 [rest of message]

Hope you are free. Send me a reply back when you get a chance.
 Joe.

·
250 OK

2. Simple Mail Transfer Protocol (4/)

SMTP Limitations

- Body must be encoded in 7 bit ASCII: no binaries
 - 7 bit = 128 possible characters
 - Images, video, etc, are inherently 8 bit
- No National Language Characters (8-bit)
- SMTP servers may have arbitrary size limit
- SMTP gateways have inconsistent ASCII to EBCDIC Mappings
- SMTP gateways to X.400 cannot handle non-textual data
 - X.400 is the messaging (notably e-mail) standard specified by the ITU-TS (International Telecommunications Union - Telecommunication Standard Sector).

2. Simple Mail Transfer Protocol (5/)

Because of limitations

- it is usually used with one of two other protocols, POP3 or IMAP that let the user save messages in a server mailbox and download them periodically from the server
- users typically use a program that uses SMTP for sending e-mail and either POP3 or IMAP for receiving e-mail
 - On Unix-based systems, sendmail is the most widely-used SMTP server for e-mail

POP3

- POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail.
- POP3 is a client/server protocol in which e-mail is received and held for you by your Internet server
- POP3 is designed to delete mail on the server as soon as the user has downloaded it
- some implementations allow users or an administrator to specify that mail be saved for some period of time

IMAP

- IMAP (Internet Message Access Protocol) is a standard protocol for accessing e-mail from your local server
- IMAP (the latest version is IMAP Version 4) is a client/server protocol in which e-mail is received and held for you by your Internet server.
- You (or your e-mail client) can view just the heading and the sender of the letter and then decide whether to download the mail.
- You can also create and manipulate multiple folders or mailboxes on the server, delete messages, or search for certain parts or an entire note.
- IMAP requires continual access to the server during the time that you are working with your mail.

POP3 vs IMAP

- less sophisticated protocol is Post Office Protocol 3 (POP3).
- With POP3, your mail is saved for you in a single mailbox on the server. When you read your mail, all of it is immediately downloaded to your computer and, except when previously arranged, no longer maintained on the server.
- IMAP provides the user more capabilities for retaining e-mail on the server and for organizing it in folders on the server.
- IMAP can be thought of as a remote file server. POP3 can be thought of as a "store-and-forward" service.

Spoof SMTP mail on mail server (6/)

C:\>telnet mail.hust.edu.vn 25

Connected to whatever.com.

Escape character is '^['.

220 whatever.com ESMTP Exim 4.20 Tue, 15 Jul 2003 14:09:37-0700

MAIL FROM: <phanliem@soict.hust.edu.vn>

250 OK

RCPT TO: <admint@hust.edu.vn>

250 Accepted

DATA

354 Enter message, ending with "." on a line by itself

Subject: a message from God

Don't fake any more email or I'll send you to hell.

.

250 OK id=19cX3k-0000Cw-Mm

QUIT

221 whatever.com closing connection

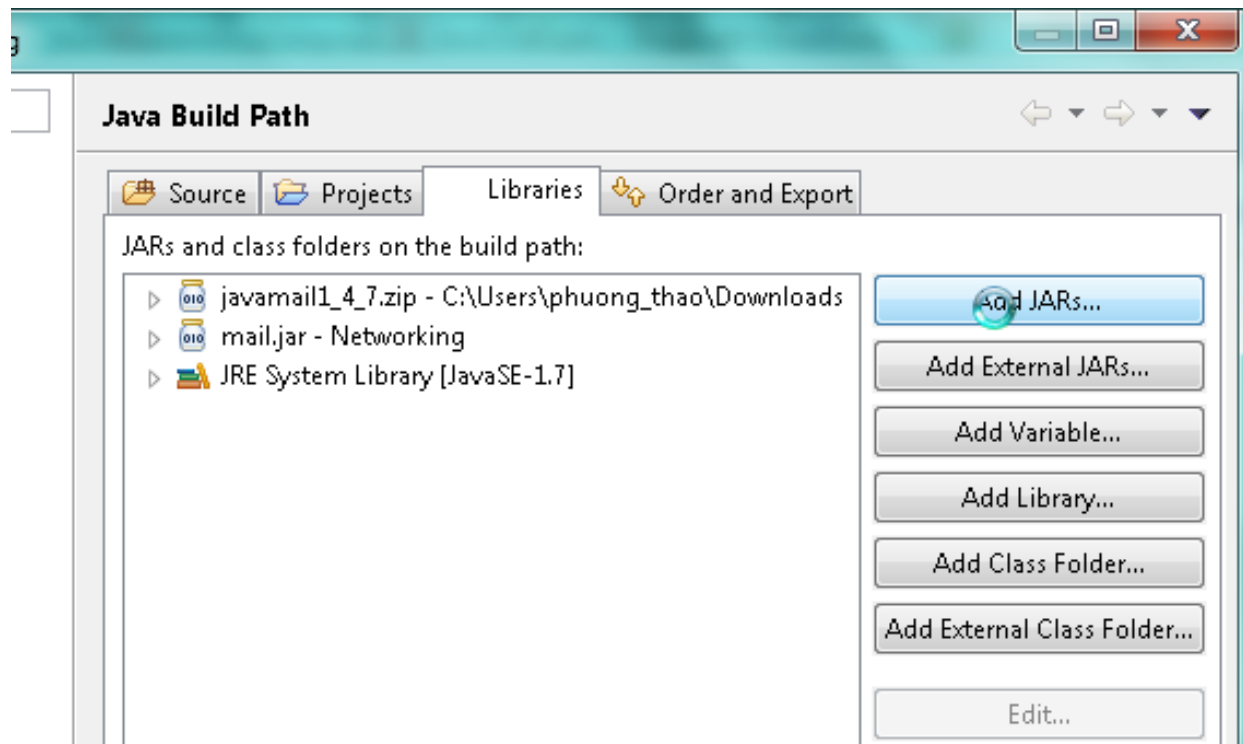
Connection closed by foreign host.

3. Send & receive email

- Tham khảo:
- <http://buddhimawijeweera.wordpress.com/2011/02/09/send-receiveemailsjava/>

JavaMail package

- http://download.oracle.com/otn-pub/java/javamail/1.4.7/javamail1_4_7.zip
- Giải nén, copy mail.jar vào thư mục của project
- Properties/Libraries/Add JARs




```
import java.util.*;
import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.Message.RecipientType;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.Store;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.swing.JOptionPane;

public class GmailClient {

    private String userName;
    private String password;
    private String sendingHost;
    private int sendingPort;
    private String from;
    private String to;
    private String subject;
    private String text;
    private String receivingHost;
    //    private int receivingPort;

    public void setAccountDetails(String userName,String password){

        this.userName=userName;//sender's email can also use as User Name
        this.password=password;

    }
```

```

public void sendGmail(String from, String to, String subject, String text){
    // This will send mail from -->sender@gmail.com to -->receiver@gmail.com
    this.from=from;
    this.to=to;
    this.subject=subject;
    this.text=text;

    // For a Gmail account--sending mails-- host and port should be as follows
    this.sendingHost="smtp.gmail.com";
    this.sendingPort=465;
    Properties props = new Properties();
    props.put("mail.smtp.host", this.sendingHost);
    props.put("mail.smtp.port", String.valueOf(this.sendingPort));
    props.put("mail.smtp.user", this.userName);
    props.put("mail.smtp.password", this.password);
    props.put("mail.smtp.auth", "true");
    Session session1 = Session.getDefaultInstance(props);
    Message simpleMessage = new MimeMessage(session1);

    //MIME stands for Multipurpose Internet Mail Extensions
    InternetAddress fromAddress = null;
    InternetAddress toAddress = null;
    try {

        fromAddress = new InternetAddress(this.from);
        toAddress = new InternetAddress(this.to);
    } catch (AddressException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Sending email to: " +
            to + " failed !!!", "Failed to Send!!!", JOptionPane.ERROR_MESSAGE);
    }
}

```

```

try {
    simpleMessage.setFrom(fromAddress);
    simpleMessage.setRecipient(RecipientType.TO, toAddress);
    // to add CC or BCC use
    // simpleMessage.setRecipient(RecipientType.CC,
    //     new InternetAddress("CC_Recipient@any_mail.com"));
    // simpleMessage.setRecipient(RecipientType.BCC,
    //     new InternetAddress("CBC_Recipient@any_mail.com"));

    simpleMessage.setSubject(this.subject);
    simpleMessage.setText(this.text);
    //sometimes Transport.send(simpleMessage); is used, but for gmail it's different

    Transport transport = session1.getTransport("smtps");
    transport.connect (this.sendingHost,sendingPort, this.userName, this.password);
    transport.sendMessage(simpleMessage, simpleMessage.getAllRecipients());
    transport.close();
    JOptionPane.showMessageDialog(null, "Mail sent successfully ...","Mail sent",
        JOptionPane.PLAIN_MESSAGE);
} catch (MessagingException e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(null, "Sending email to: " +
        to + " failed !!!",
        "Failed to Send!!!", JOptionPane.ERROR_MESSAGE);
}
}

```

```

public void readGmail(){
    /*this will print subject of all messages in the inbox of sender@gmail.com*/
    this.receivingHost="imap.gmail.com";//for imap protocol
    Properties props2=System.getProperties();
    // I used imaps protocol here
    props2.setProperty("mail.store.protocol", "imaps");

    Session session2=Session.getDefaultInstance(props2, null);
    try {
        Store store=session2.getStore("imaps");
        store.connect(this.receivingHost,this.userName, this.password);
        Folder folder=store.getFolder("INBOX");//get inbox
        folder.open(Folder.READ_ONLY);//open folder only to read
        Message message[]=folder.getMessages();
        for(int i=0;i<message.length;i++){
            //print subjects of all mails in the inbox
            System.out.println(message[i].getSubject());
            //anything else you want
        }
        //close connections
        folder.close(true);
        store.close();
    } catch (Exception e) {
        System.out.println(e.toString());
    }
}

```

```
public static void main(String[] args) {  
  
    //Sender must be a Gmail Account holder  
    String mailFrom=new String("liempt.vsec@gmail.com");  
  
    //but here you can send to any type of mail account  
    String mailTo=new String("phanliem@soict.hust.edu.vn");  
  
    String senderPassword=new String("gmail");  
  
    String senderUserName=new String("liempt.vsec");  
  
    //Mention your email subject and content  
    String mailSubject=new String("Testing Mail");  
    String mailText=new String("Have an Nice Day .....!!!");  
  
    //Create a GmailClient object  
    GmailClient newGmailClient=new GmailClient();  
  
    //Setting up account details  
    newGmailClient.setAccountDetails(senderUserName, senderPassword);  
  
    //Send mail  
    newGmailClient.sendGmail(mailFrom, mailTo, mailSubject, mailText);  
  
    //Receive mails  
    newGmailClient.readGmail();  
}  
  
}
```