



LẬP TRÌNH JAVA 3

BÀI 8: NETWORKING, DOWNLOAD, EMAIL

PHẦN 1

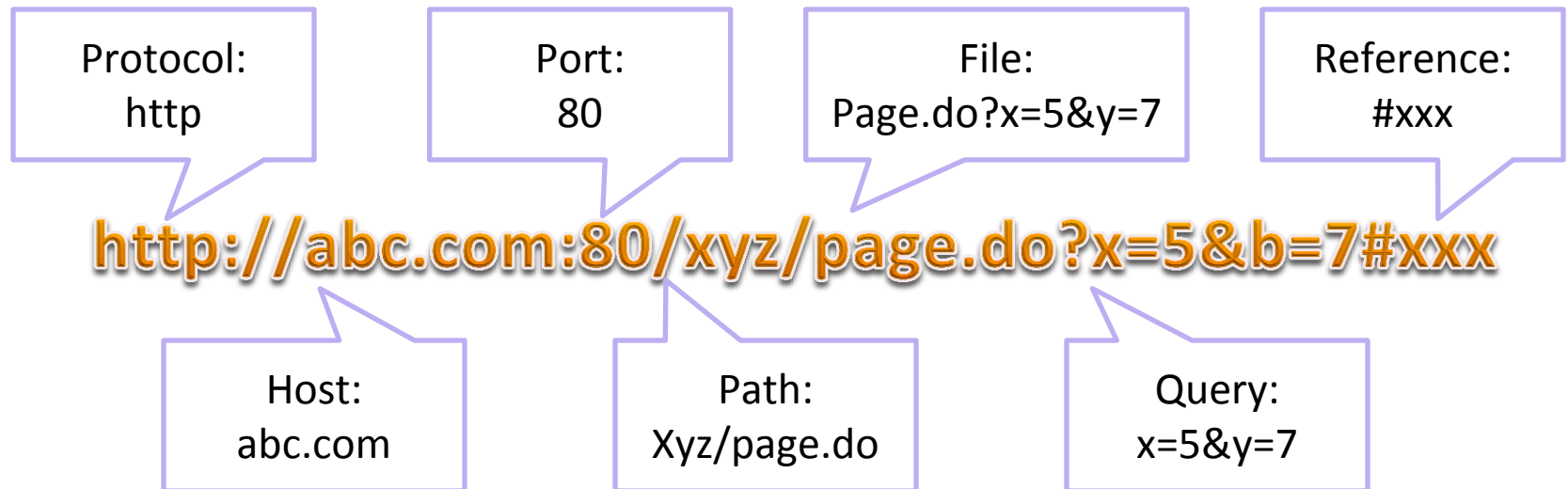
- ◎ Kết thúc bài học này bạn có khả năng
 - ❖ Download tài nguyên mạng
 - ❖ Gửi email thông qua gmail



- ❑ Tài nguyên trên mạng (trang web, hình ảnh, video...) có thể truy cập thông qua địa chỉ URL



- ❑ Java cung cấp lớp `java.net.URL` để có thể gửi/nhận dữ liệu với các URL đó.



Kiểu trả về	Phương thức	Mô tả
String	<code>getFile()</code>	Lấy tên file
String	<code>getHost()</code>	Lấy tên server
String	<code>getPath()</code>	Lấy đường dẫn
int	<code>getPort()</code>	Lấy cổng
String	<code>getProtocol()</code>	Lấy giao thức
String	<code>getQuery()</code>	Lấy chuỗi truy vấn
String	<code>getRef()</code>	Lấy tham chiếu trong trang
URLConnection	<code>openConnection()</code>	Mở kết nối đến URL
InputStream	<code>openStream()</code>	Lấy luồng dữ liệu từ tài nguyên

```
try {
    URL url = new URL("http://abc.com:80/xyz/page.jsp?x=5&y=7#xxx");
    System.out.println("Protocol: " + url.getProtocol());
    System.out.println("Host: " + url.getHost());
    System.out.println("Port: " + url.getPort());
    System.out.println("Path: " + url.getPath());
    System.out.println("File: " + url.getFile());
    System.out.println("Query: " + url.getQuery());
    System.out.println("Ref: " + url.getRef());
}
catch (MalformedURLException ex) {
    ex.printStackTrace();
}
```

Protocol: http
 Host: abc.com
 Port: 80
 Path: /xyz/page.jsp
 File: /xyz/page.jsp?x=5&y=7
 Query: x=5&y=7
 Ref: xxx

```
ByteArrayOutputStream buffer = new ByteArrayOutputStream();
```

```
URL url = new URL("http://www.vnexpress.net");
```

```
InputStream is = url.openStream();
```

```
byte[] block = new byte[4*1024]; // 4KB
```

```
while(true){
```

```
    int n = is.read(block);
```

```
    if(n <= 0){
```

```
        break;
```

```
    }
```

```
        buffer.write(block, 0, n);
```

```
    }
```

```
is.close();
```

```
byte[] data = buffer.toByteArray();
```

```
buffer.close();
```

Chứa dữ liệu download

Mở luồng dữ liệu đến tài nguyên

Chỉ nên đọc 1 lần tối đa 4KB

Dừng khi đã hết dữ liệu

Tích lũy dữ liệu đọc được

Lấy nội dung trang web

❑ Thiết kế form

The screenshot displays a Java IDE interface with the following components:

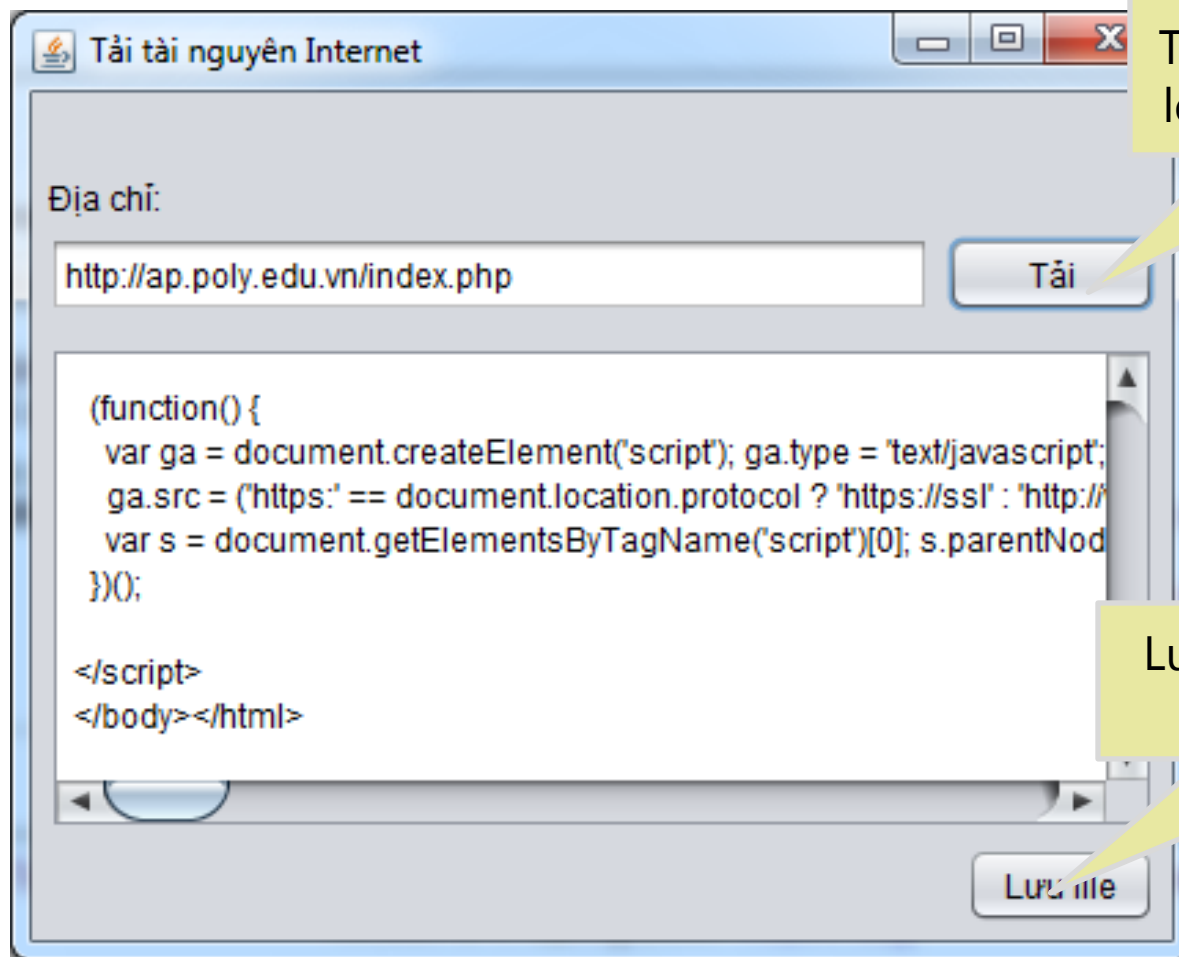
- Projects Panel:** Shows a project structure with folders 'Slide6' and 'Slide8', and a file 'GetSourceInternet.java'.
- Other Components - Navigator:** A tree view of available UI components. The 'Other Components' folder is expanded, showing a list of components including:
 - [JFrame]
 - label jLabel1 [JLabel]
 - txtUrl [JTextField]
 - btnTai [JButton]
 - jScrollPane1 [JScrollPane]
 - txtNoiDung [JTextArea]
 - btnLuu [JButton]This list is highlighted with a red rectangular box.
- Design View:** The main workspace shows the visual design of the 'GetSourceInternet.java' form. It includes:
 - A label 'Địa chỉ:' (Address:).
 - A text input field for the address.
 - A button labeled 'Tải' (Load).
 - A large text area for content.
 - A button labeled 'Lưu file' (Save file) at the bottom right.

□ Viết code cho button “Tải”

```
private void btnTaiActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        ByteArrayOutputStream buffer = new ByteArrayOutputStream();  
        URL url = new URL(txtUrl.getText());  
        InputStream is = url.openStream();  
        byte[] block = new byte[4*1024]; // 4KB  
        while(true){  
            int n = is.read(block);  
            if(n <= 0){  
                break;  
            }  
            buffer.write(block, 0, n);  
        }  
        is.close();  
        txtNoiDung.setText(buffer.toString("utf-8"));  
        buffer.close();  
    } catch (Exception ex) {  
        System.out.println(ex);  
    }  
}
```

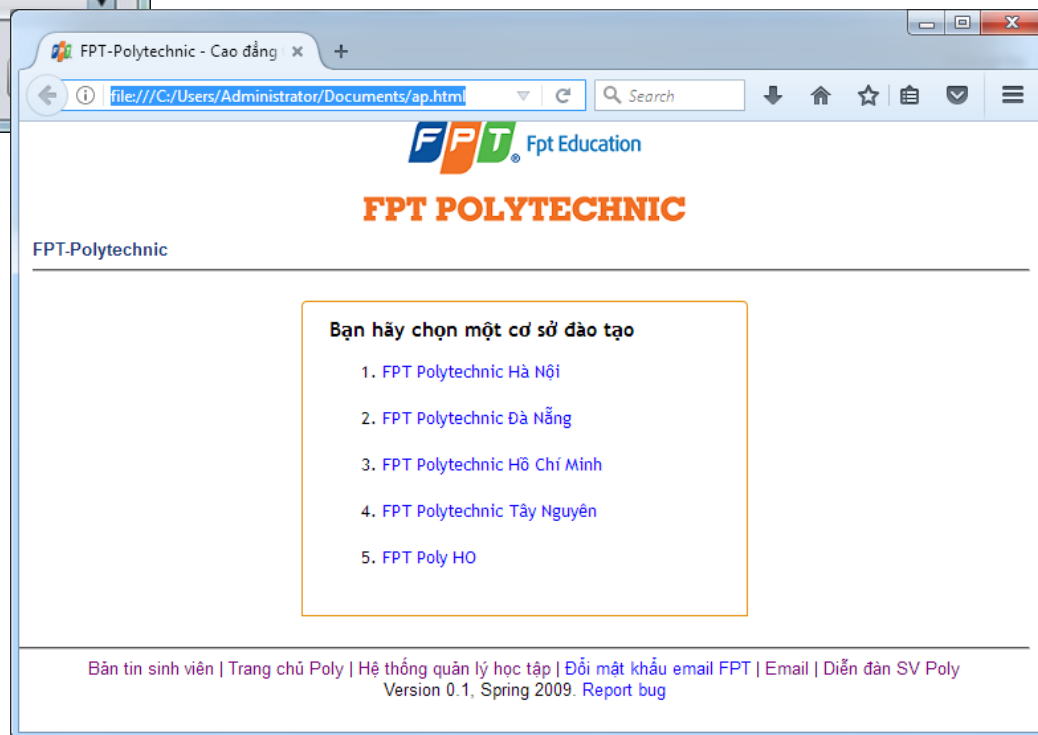
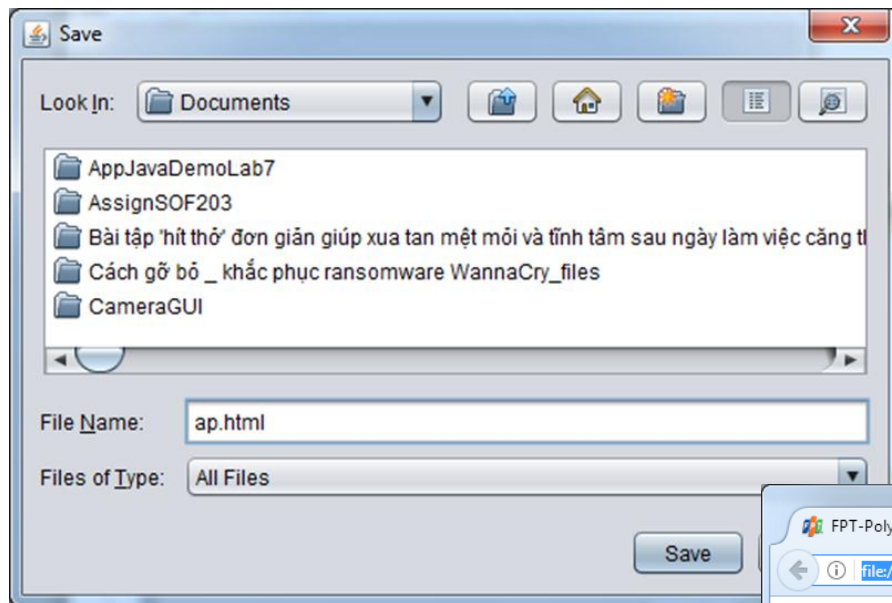
❑ Viết code cho button “Lưu file”

```
private void btnLuuActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        JFileChooser filechoo = new JFileChooser();  
        if(filechoo.showSaveDialog(null) == JFileChooser.APPROVE_OPTION) {  
            File file = filechoo.getSelectedFile();  
            FileOutputStream fos = new FileOutputStream(file);  
            URL url = new URL(txtUrl.getText());  
            InputStream is = url.openStream();  
            byte[] block = new byte[4*1024]; // 4KB  
            while(true) {  
                int n = is.read(block);  
                if(n <= 0) {  
                    break;  
                }  
                fos.write(block, 0, n);  
            }  
            is.close();  
            fos.close();  
        }  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
}
```



Tải và hiển thị trang web
lên ô nội dung bên dưới

Lưu nội dung vào file được
nhập từ JFileChooser





DEMO

Chạy và giải thích



- ❑ Gửi email là hoạt động thường xuyên xảy ra của 1 ứng dụng. Có thể người dùng gửi cho 1 ai đó, hoặc có thể là thông báo của hệ thống
- ❑ Để gửi email trong Java bạn cần 2 thư viện
 - ❖ Mail.jar
 - ❖ Activation.jar



❑ Các phương thức được sử dụng của MimeMessage trong việc gửi email.

Phương thức	Mô tả
setFrom(InternetAddress)	Cung cấp địa chỉ email người gửi
setReplyTo(InternetAddress[])	Cung cấp địa chỉ email người nhận phản hồi
addRecipients(RecipientType, Address[])	Cung cấp danh sách địa chỉ email người cùng nhận
setSubject(String)	Cung cấp tiêu đề mail
setContent(String, String)	Cung cấp nội dung mail
setSentDate(Date)	Cung cấp ngày gửi mail

Properties

Chuẩn bị thông số cấu hình

Authenticator

Cung cấp tài khoản kết nối mail server

Session

Mở một session dựa vào cấu hình và authenticator ở trên

MimeMessage

Xây dựng mail: người nhận, người gửi, tiêu đề, nội dung...

Transport

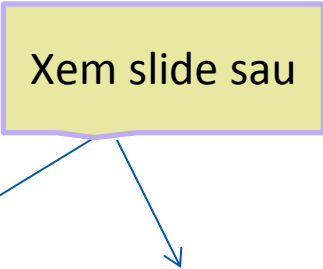
Gửi mail


```
try{
    String fromName = "Nguyễn Văn Sender";
    String fromEmail = "sender@gmail.com";
    String toEmails = "a@gmail.com,b@fpt.edu.vn";
    String subject = "Tiêu đề mail";
    String body = "Nội dung mail";

    Session session = Session.getInstance(config, authenticator);
    MimeMessage mail = new MimeMessage(session);

    InternetAddress sender = new InternetAddress(fromEmail, fromName, "utf-8");
    mail.setFrom(sender);
    mail.setReplyTo(new InternetAddress[]{sender});
    mail.addRecipients(Message.RecipientType.TO, toEmails);
    mail.setSubject(subject, "utf-8");
    mail.setContent(body, "text/html; charset=utf-8");
    mail.setSentDate(new Date());

    Transport.send(mail);
}
catch (Exception e) {
    throw new RuntimeException(e);
}
```



CODE GỬI EMAIL QUA GMAIL

```
// Các thông số gmail
Properties config = new Properties();
config.setProperty("mail.smtp.host", "smtp.gmail.com");
config.setProperty("mail.smtp.port", "465");
config.setProperty("mail.smtp.starttls.enable", "true");
config.setProperty("mail.smtp.auth", "true");
config.setProperty("mail.smtp.socketFactory.class",
    "javax.net.ssl.SSLSocketFactory");
config.setProperty("mail.smtp.socketFactory.fallback", "false");

// Đăng nhập gmail
Authenticator authenticator = new Authenticator() {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        String email = "user@gmail.com";
        String password = "songlong";
        return new PasswordAuthentication(email, password);
    }
};
```

❑ Viết code cho button “Send”

The screenshot displays an IDE environment for a Java Swing application titled "Send Mail".

Left Panel (Other Components - Navigator):

- Projects: SendMail.java
- Test Packages
- Libraries: sqljdbc4.jar, javax.mail.jar
- Other Components - Navigator:
 - Form SendMail
 - Other Components
 - JFrame
 - jPanel1 [JPanel]
 - label jLabel1 [JLabel]
 - txtUser [JTextField]
 - label jLabel2 [JLabel]
 - txtPass [JPasswordField]
 - jPanel2 [JPanel]
 - label jLabel3 [JLabel]
 - label jLabel4 [JLabel]
 - label jLabel5 [JLabel]
 - txtTo [JTextField]
 - txtSubject [JTextField]
 - jScrollPane1 [JScrollPane]
 - txtMessage [JTextArea]
 - btnSend [JButton]
 - label jLabel6 [JLabel]

Right Panel (Design View):

The design view shows a window titled "Send Mail" with two main sections:

- Send Section:**
 - Username: [Text Field]
 - Password: [Text Field]
- Receive Section:**
 - To: [Text Field]
 - Subject: [Text Field]
 - Message: [Text Area]
 - Send [Button]

❑ Thiết kế form như hình

```
private void btnSendActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Properties p = new Properties();
        p.put("mail.smtp.auth", "true");
        p.put("mail.smtp.starttls.enable", "true");
        p.put("mail.smtp.host", "smtp.gmail.com");
        p.put("mail.smtp.port", 587);
        String accountName = txtUser.getText();
        String accountPassword = txtPass.getText();
        Session s = Session.getInstance(p,
            new javax.mail.Authenticator() {
                protected PasswordAuthentication getPasswordAuthentication() {
                    return new PasswordAuthentication(accountName, accountPassword);
                }
            });
        String from = txtUser.getText();
        String to = txtTo.getText();
        String subject = txtSubject.getText();
        String body = txtMessage.getText();
        Message msg = new MimeMessage(s);
        msg.setFrom(new InternetAddress(from));
        msg.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));
        msg.setSubject(subject);
        msg.setText(body);
        Transport.send(msg);
        JOptionPane.showMessageDialog(null, "Mail was sent successfully.", "Message",
            javax.swing.JOptionPane.INFORMATION_MESSAGE);
    } catch (MessagingException ex) {
        Logger.getLogger(SendMail.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```



Send Mail

Send

Username:

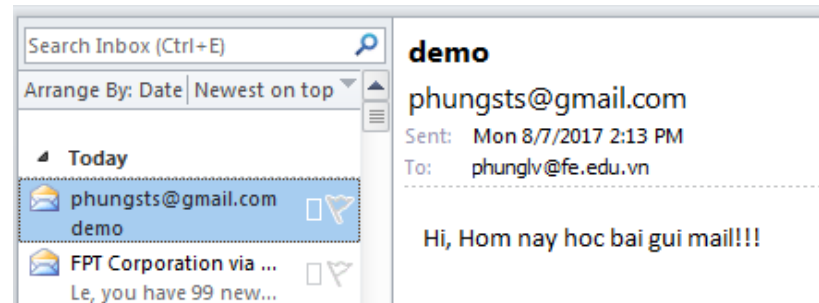
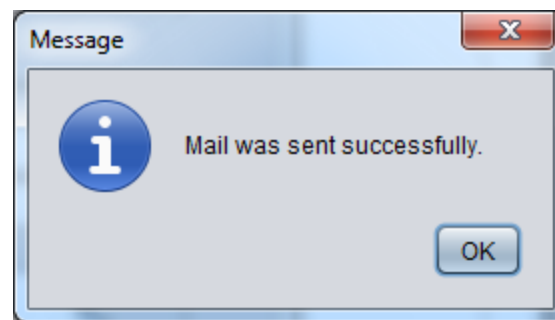
Password:

Receive

To:

Subject:

Message:





LẬP TRÌNH JAVA 3

BÀI 8: NETWORKING, DOWNLOAD, EMAIL

PHẦN 2

- ❑ Để gửi mail cho nhiều người cùng nhận bạn chỉ cần bổ sung các email ấy vào CC hoặc BCC.
- ❑ Sự khác biệt giữa CC và BCC là CC là những email sẽ được người nhận nhìn thấy trên mail còn BCC thì không.
- ❑ Mã bổ sung email cùng nhận:

```
String ccEmails = "cc1@gmail.com,cc2@yahoo.com,cc3@hotmail.com";  
mail.addRecipients(Message.RecipientType.CC, ccEmails);
```

```
String bccEmails = "bcc1@gmail.com,bcc2@yahoo.com";  
mail.addRecipients(Message.RecipientType.BCC, bccEmails);
```

```
Transport.send(mail);
```

- ☐ Bài tập: Nâng cấp bài Send Email trên có thêm chức năng CC hoặc BCC


```
// Phần nội dung mail chính
MimeBodyPart contentPart = new MimeBodyPart();
contentPart.setContent(body, "text/html; charset=utf-8");

// Phần file đính kèm
MimeBodyPart filePart = new MimeBodyPart();
File file = new File("c:/temp/a.gif");
FileDataSource fds = new FileDataSource(file);
filePart.setDataHandler(new DataHandler(fds));
filePart.setFileName(file.getName());

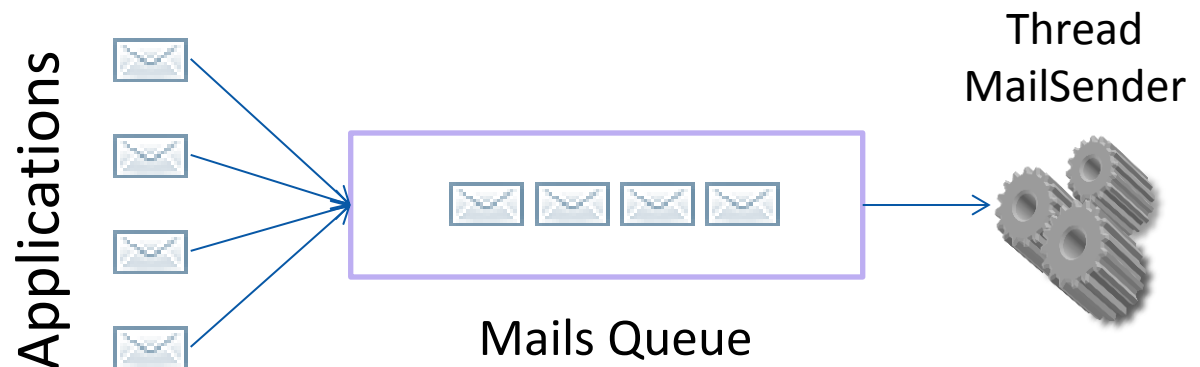
// Bổ sung các phần vào multi-part
MimeMultipart multiPart = new MimeMultipart();
multiPart.addBodyPart(contentPart);
multiPart.addBodyPart(filePart);

// Thiết lập nội dung mail là multi-part
mail.setContent(multiPart);

Transport.send(mail);
```

- ☐ Bài tập: Nâng cấp bài Send Email trên có thêm chức năng đính kèm theo file

- ❑ Chi phí (CPU và bộ nhớ) dành cho gửi mail là rất cao. Đôi khi hệ thống phải thực hiện gửi nhiều email một lúc dẫn đến tài nguyên cạn kiệt dễ gây tê liệt hệ thống.
- ❑ Giải pháp khắc phục là tại queue nắm giữ các mail và có 1 thread chuyển trách gửi tuần tự từng email.



```
public class MailSender extends Thread{
    static {
        MailSender sender = new MailSender();
        sender.start();
    }
    static final List<MimeMessage> queue = new ArrayList<>();

    public static void queue(MimeMessage mail) {
        synchronized(queue) {
            queue.add(mail);
            queue.notify();
        }
    }

    @Override
    public void run() { ...24 lines }
}
```

Xem slide sau

```
while (true) {
    try {
        synchronized (queue) {
            if (queue.size() > 0) {
                try {
                    MimeMessage mail = queue.remove(0);
                    Transport.send(mail);
                    System.out.println("The mail was sent.");
                }
                catch (MessagingException e) {
                    System.out.println("Unable to send mail.");
                }
            }
            else {
                queue.wait();
            }
        }
    }
    catch (InterruptedException e) {
        break;
    }
}
```

```
Session session = Session.getInstance(config, authenticator);
MimeMessage mail = new MimeMessage(session);

InternetAddress sender = new InternetAddress(fromEmail, fromName, "utf-8");
mail.setFrom(sender);
mail.setReplyTo(new InternetAddress[]{sender});
mail.addRecipients(Message.RecipientType.TO, toEmails);
mail.setSubject(subject, "utf-8");
mail.setContent(body, "text/html; charset=utf-8");
mail.setSentDate(new Date());

//Transport.send(mail);
MailSender.queue(mail);
```

Thay vì gọi ~~Transport.send(mail)~~ để gửi email thì sử dụng **MailSender.queue(mail)** để bổ sung email vào queue và sẽ được gửi bởi 1 Thread khác

- ❖ Download tài nguyên mạng
- ❖ Gửi email thông qua gmail





KẾT THÚC