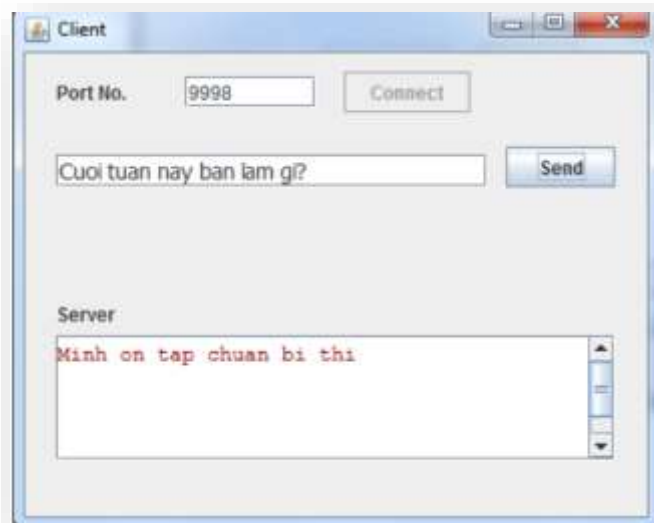
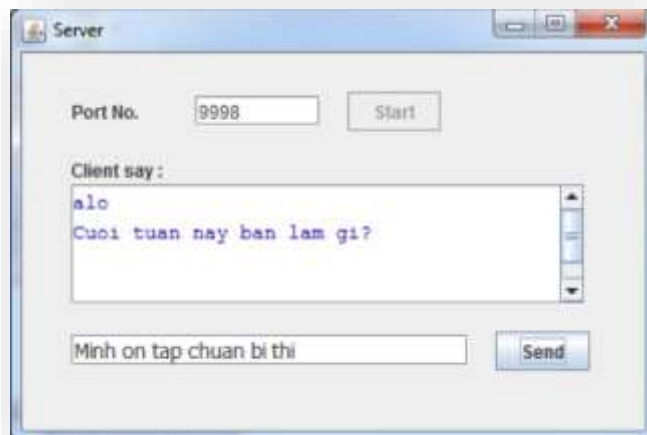


MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng:

- ✓ Hiểu về Socket, Lập trình Socket TCP
- ✓ Có khả năng tạo ra các chương trình chat
- ✓ Viết chương trình gửi email

BÀI 1: Viết chương trình chat dùng Socket (4 ĐIỂM)



Code tham khảo:

Server:

```
public void startSever(){
    try {
        btnStart.setEnabled(b, false);
        serverSocket = new ServerSocket(port: Integer.parseInt(txtPort.getText()));
        txtHistory.setText(b: "connecting.....");
        socket = serverSocket.accept();
        txtHistory.setText(b: "connected");
        Thread t = new Thread(task: FSever.this);
        t.start();
    } catch (IOException ex) {
        Logger.getLogger(name: FSever.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

```
public void sendMessage(){
    try {
        dataOutputStream = new DataOutputStream(out: socket.getOutputStream());
        dataOutputStream.writeUTF(msg: txtMessage.getText());
    } catch (Exception e) {
    }
}

@Override
public void run() {
    while (true) {
        try {
            if(socket != null){
                try {
                    dataInputStream = new DataInputStream(in: socket.getInputStream());
                    String msString = dataInputStream.readUTF();
                    System.out.println(msString);
                    txtHistory.setText(b: msString);
                } catch (IOException ex) {
                    Logger.getLogger(name: FSever.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
                }
            }
            Thread.sleep(milliseconds: 10);
        } catch (InterruptedException ex) {
            Logger.getLogger(name: FSever.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
        }
    }
}
```

Client:

```
public void connectSever(){
    txtHistory.setText(t: "connecting.....");
    btnStart.setEnabled(b: false);
    try {
        socket = new Socket(host: "localhost", port: Integer.parseInt(s: txtPort.getText()));
        txtHistory.setText(t: "connecttd");
        Thread t = new Thread(task: FClient.this);
        t.start();
    } catch (IOException ex) {
        Logger.getLogger(name: FClient.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

```
public void sendMessage(){
    try {
        dataOutputStream = new DataOutputStream(out: socket.getOutputStream());
        dataOutputStream.writeUTF(s: txtMessage.getText());
    } catch (Exception e) {
    }
}

@Override
public void run() {
    while (true) {
        try {
            if(socket != null){
                try {
                    dataInputStream = new DataInputStream(in: socket.getInputStream());
                    String msString = dataInputStream.readUTF();
                    txtHistory.setText(t: msString);
                } catch (IOException ex) {
                    Logger.getLogger(name: FSever.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
                }
            }
            Thread.sleep(millis: 10);
        } catch (InterruptedException ex) {
            Logger.getLogger(name: FSever.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
        }
    }
}
```

BÀI 2: Nâng cấp bài 1 mô phỏng các chức năng như Messenger, yahoo (1 ĐIỂM)

BÀI 3: Thiết kế ứng dụng như hình và thực hiện các chức năng theo yêu cầu sau (3 ĐIỂM)

SEND EMAIL

Sender

Email: taint.ctict@gmail.com

Password:

Receive

Email: taint76@fpt.edu.vn

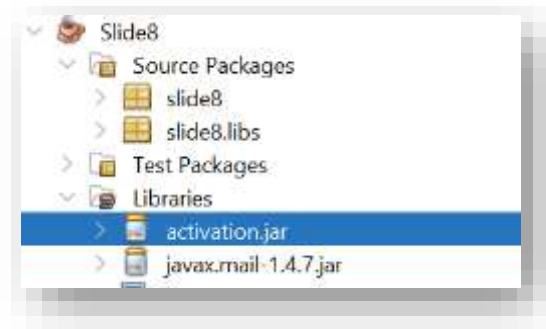
Subject: Demo Send Email

Message: Hello JAVA 2. demo send email

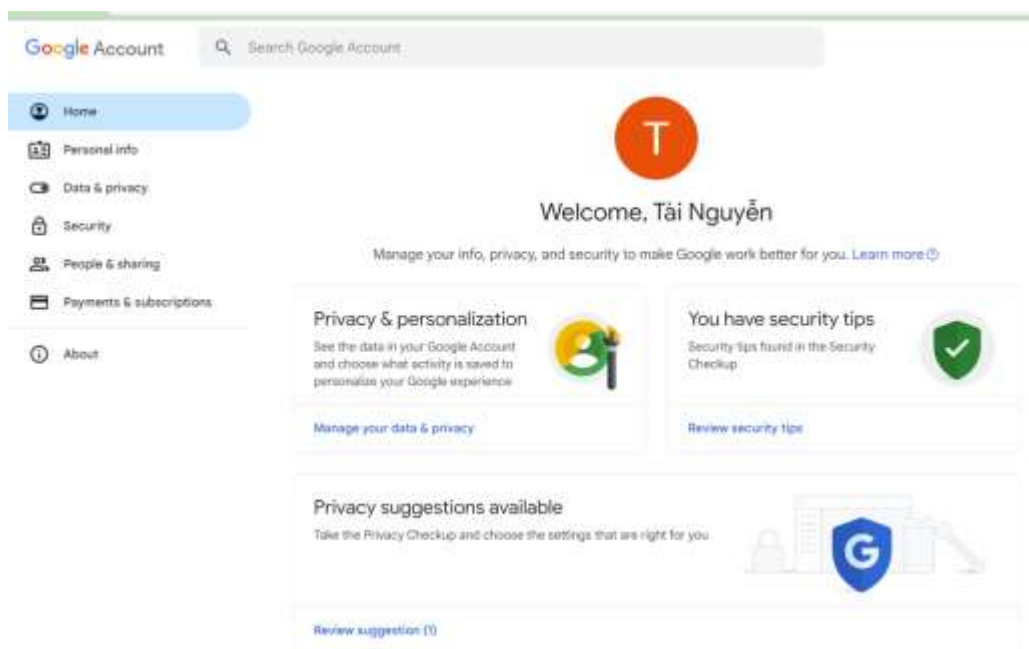
Send

Chuẩn bị:

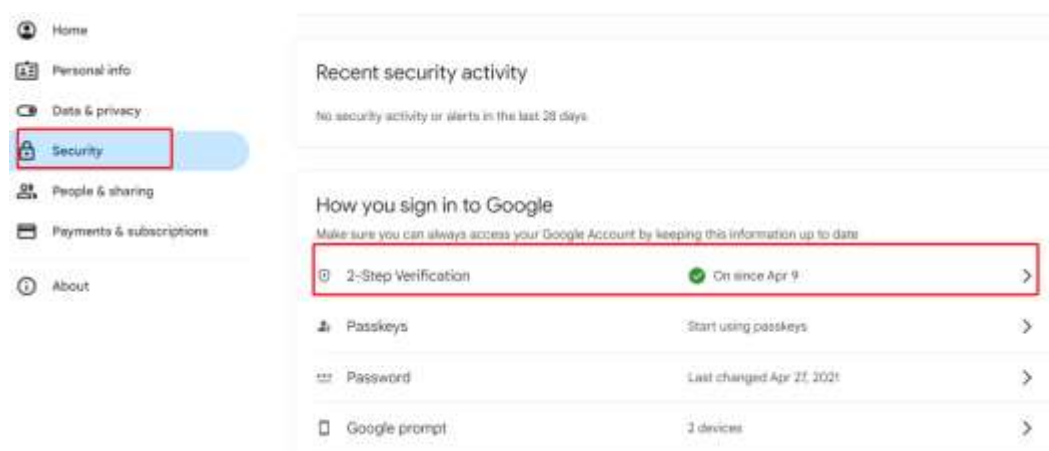
- Tải JavaMail API
- Thêm JavaMail API(javax.mail.jar) và (activation.jar) vào project



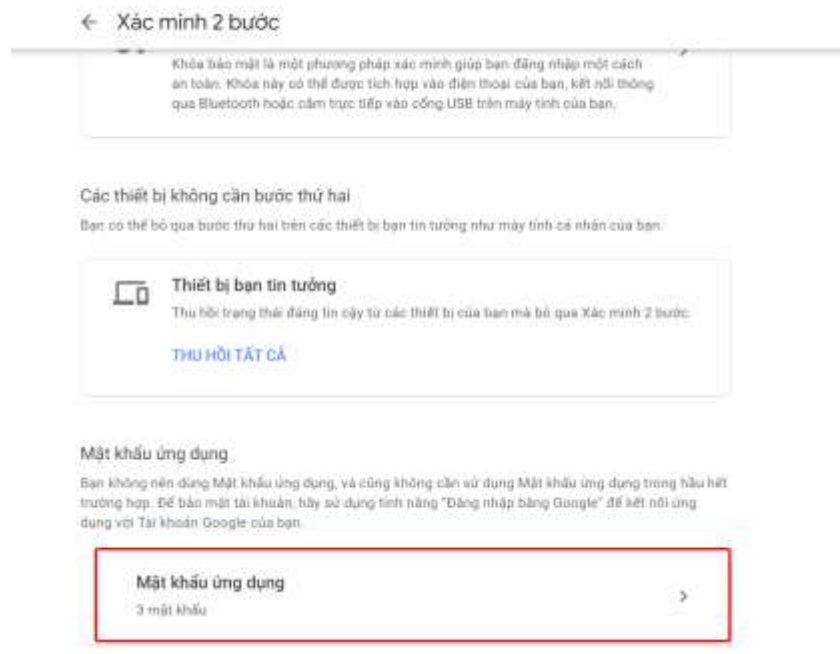
- Tài khoản gmail: Tạo mật khẩu ứng dụng cho tài khoản gmail
1. Truy cập quản lý tài khoản Google



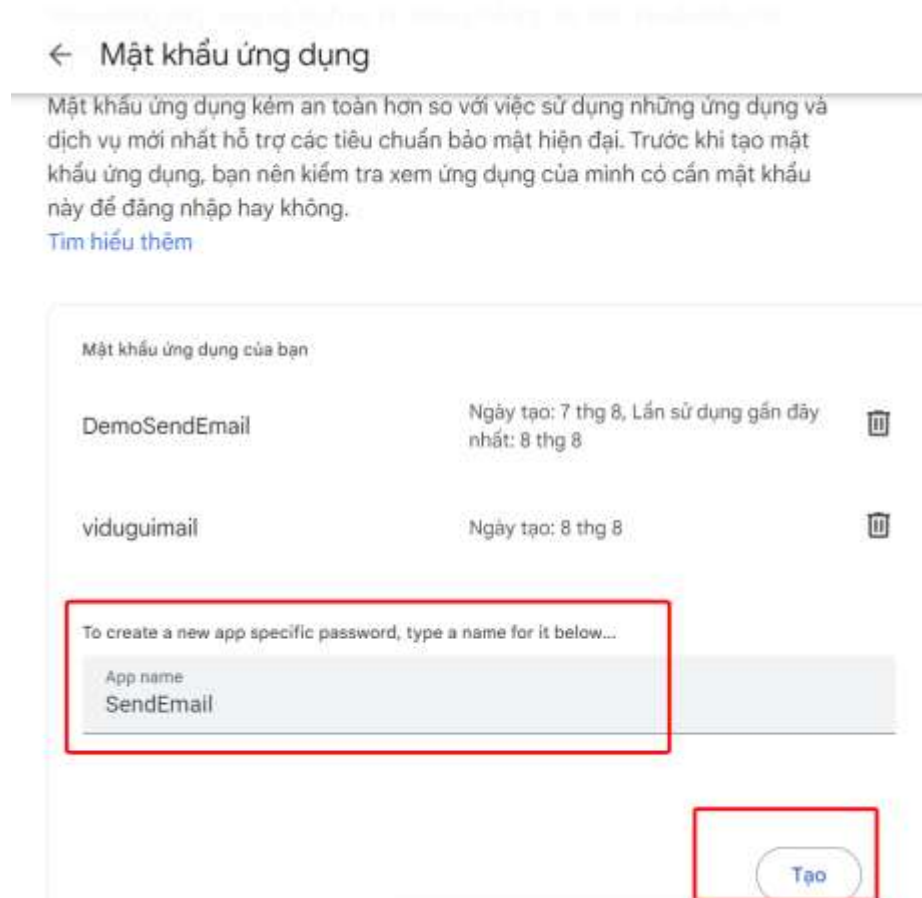
2. Chọn chế độ bảo mật “Security”. Bật “2-Step Verification



3. Di chuyển xuống cuối trang và tìm “Mật khẩu ứng dụng”



4. Nhập tên ứng dụng và bấm “Tạo”



5. Mật khẩu ứng dụng là chuỗi 16 ký tự được tạo ra



Cài đặt:

- **Bước 1:** Tạo đối tượng Properties và chỉ định thông tin host, port

```
//các thông số gmail
Properties prop = new Properties();
prop.put(key: "mail.smtp.auth", value: "true");
prop.put(key: "mail.smtp.starttls.enable", value: "true"); //TLS
prop.put(key: "mail.smtp.host", value: "smtp.gmail.com");
prop.put(key: "mail.smtp.port", value: "587");
prop.put(key: "mail.smtp.socketFactory.port", value: "587");
prop.put(key: "mail.smtp.socketFactory.class", value: "javax.net.ssl.SSLSocketFactory");
prop.put(key: "mail.smtp.ssl.protocols", value: "TLSv1.2");
```

- **Bước 2:** Tạo đối tượng Session (phiên làm việc)

```
Session session = Session.getInstance(prop,
    new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(userName, senderEmail, password);
        }
    });
```

Trong đó

senderEmail: là tài khoản gmail.

password: là mật khẩu ứng dụng được tạo ra trước đó

- **Bước 3:** Tạo đối tượng message

```
Message message = new MimeMessage(session);
message.setFrom(new InternetAddress(address:senderEmail));
message.setRecipients(
    new Message.RecipientType.TO,
    address: InternetAddress.parse(address:txtReceiveEmail.getText())
);
message.setSubject(subject: txtSubject.getText());
message.setContent(txtMessage.getText(), mimeType: "text/html; charset=utf-8");
```

- **Bước 4:** Gửi mail

```
Transport.send(msg:message);
System.out.println(x: "Gửi thành công");
```

Code tham khảo

```
public void sendEmail(){
    //các thông số gmail
    Properties prop = new Properties();
    prop.put("mail.smtp.auth", "true");
    prop.put("mail.smtp.starttls.enable", "true"); //TLS
    prop.put("mail.smtp.host", "smtp.gmail.com");
    prop.put("mail.smtp.port", "587");
    prop.put("mail.smtp.socketFactory.port", "587");
    prop.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    prop.put("mail.smtp.ssl.protocols", "TLSv1.2");
    //sử dụng qua gmail
    String senderEmail = txtSenderEmail.getText();
    String password = new String(txtPass.getPassword());
    Session session = Session.getInstance(prop,
        new javax.mail.Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(senderEmail, password);
            }
        });
    try {
        Message message = new MimeMessage(session);
        message.setFrom(new InternetAddress(senderEmail));
        message.setRecipients(
            new Message.RecipientType.TO,
            InternetAddress.parse(txtReceiveEmail.getText())
        );
        message.setSubject(txtSubject.getText());
        message.setContent(txtMessage.getText(), mimeType: "text/html; charset=utf-8");
        Transport.send(message);
        System.out.println("Gửi thành công");
    } catch (MessagingException e) {
        e.printStackTrace();
    }
}
```

BÀI 4: Phát triển thêm các chức năng từ bài 3 (2 ĐIỂM)

- **Attach file:** cho gửi kèm theo file
- **CC:** cho phép cùng lúc nội dung email được gửi đến nhiều người

*** YÊU CẦU NỘP BÀI:

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết ---