TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ MÔN LẬP TRÌNH MẠNG CƠ BẢN**

**XÂY DỰNG CÁC ỨNG DỤNG TRAO ĐỔI THÔNG TIN GIỮA CÁC MÁY TÍNH SỬ DỤNG GIAO THỨC TCP VÀ UDP**

*Người hướng dẫn*: **T.S. BÙI QUY ANH**

*Người thực hiện*: **Phạm Huỳnh Anh Thư -** *MSSV:* **52000409**

*Lớp:* **20050401 -** *Khoá:* **K24**

**Đỗ Nhựt Khả Vy -** *MSSV:* **52100513**

*Lớp:* **21050401 -** *Khoá:* **K25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ MÔN LẬP TRÌNH MẠNG CƠ BẢN**

**XÂY DỰNG CÁC ỨNG DỤNG TRAO ĐỔI THÔNG TIN GIỮA CÁC MÁY TÍNH SỬ DỤNG GIAO THỨC TCP VÀ UDP**

*Người hướng dẫn*: **T.S. BÙI QUY ANH**

*Người thực hiện*: **Phạm Huỳnh Anh Thư -** *MSSV:* **52000409**

*Lớp:* **20050401 -** *Khoá:* **K24**

**Đỗ Nhựt Khả Vy -** *MSSV:* **52100513**

*Lớp:* **21050401 -** *Khoá:* **K25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Tôn Đức Thắng đã đưa môn lập trình mạng căn bản vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn chân thành đến giảng viên bộ môn – Th.S Bùi Quy Anh, đã dạy dỗ và tâm huyết truyền đạt những kiến thức quý giá cho em trong suốt thời gian học tập vừa qua. Đồng thời, thầy đã mở rộng vốn kiến thức của em từ lý thuyết để áp dụng vào những vấn đề thực tế trong chuyên ngành, bên cạnh đó là những lời định hướng hết sức thiết thực cho tương lai của em. Cuối cùng em xin chân thành cảm ơn quý thầy cô ở khoa Công nghệ thông tin đã cất công tạo điều kiện giảng dạy môn học này qua các tài liệu, đề cương được chuẩn bị hết sức chu đáo đầy đủ. Môn học hệ cơ sở dữ liệu là môn học thú vị, bổ ích và có tính thực tế cao. Đảm bảo cung cấp đầy đủ kiến thức, kỹ năng, giúp sinh viên có thể ứng dụng vào thực tế. Tuy nhiên, do khả năng tiếp thu thực tế còn nhiều hạn hẹp, kiến thức chưa sâu rộng. Mặc dù bản thân đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó tránh khỏi những thiếu sót, kính mong quý thầy cô xem xét và góp ý để bài báo cáo của em được hoàn thiện và tốt hơn.

Em xin chân thành cảm ơn!

**BÁO CÁO ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Th.S Bùi Quy Anh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 10 tháng 12 năm 2022*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phạm Huỳnh Anh Thư*

*Đỗ Nhựt Khả Vy*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Bài báo cáo cuối kỳ được thực hiện nhằm đúc kết môn học, ứng dụng được những kiến thức vừa qua và xây dựng mô phỏng ứng dụng cơ bản về việc trao đổi thông tin bằng giao thức UDP và TCP.

* Hiểu và ứng dụng UDP để xây dựng nên mô hình chat giữa server và client đơn giản, qua đó thấy được những đặc điểm của giao thức này.
* Hiểu và ứng dụng giao thức TCP:

+ Xây dựng mô hình chat server – client đơn giản.

+ Chương trình từ điển.

+ Chương trính máy tính.

+ Chương trình tính diện tích hình học.

+ Chương trình giải hệ phương trình bậc I 2 ẩn.

+ Chương trình truyền tin FTP.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc121601715)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc121601716)

[TÓM TẮT iv](#_Toc121601717)

[MỤC LỤC 1](#_Toc121601718)

[CHƯƠNG 1 – PHÂN TÍCH YÊU CẦU 3](#_Toc121601719)

[1.1 Tổng quan chương tình 3](#_Toc121601720)

[1.2 Giải thích chương trình 3](#_Toc121601721)

[CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT 4](#_Toc121601722)

[CHƯƠNG 3 – CHƯƠNG TRÌNH DEMO 6](#_Toc121601723)

[TỔNG QUÁT CHƯƠNG TRÌNH (MAIN MENU ) 6](#_Toc121601724)

[3.1 CHƯƠNG TÌNH CHAT SỬ DỤNG GIAO THỨC TCP 6](#_Toc121601725)

[3.1.1 Giới thiệu chương trình 6](#_Toc121601726)

[3.1.2 Phân tích biểu đồ 7](#_Toc121601727)

[3.1.3 Giao diện chương trình 8](#_Toc121601728)

[3.1.4 Code chương trình 8](#_Toc121601729)

[3.2 CHƯƠNG TRÌNH CHAT SỬ DỤNG GIAO THỨC UDP 19](#_Toc121601730)

[3.1.1 Giới thiệu chương trình 19](#_Toc121601731)

[3.1.2 Phân tích biểu đồ 19](#_Toc121601732)

[3.1.3 Giao diện chương trình 19](#_Toc121601733)

[3.1.4 Code chương trình 20](#_Toc121601734)

[3.3 CÁC CHƯƠNG TRINHG ỨNG DỤNG KHÁC 39](#_Toc121601735)

[3.3.1 CHƯƠNG TÌNH TỪ ĐIỂN 39](#_Toc121601736)

[3.3.2 CHƯƠNG TRÌNH TÍNH TIỆN TÍCH HÌNH HOC 49](#_Toc121601737)

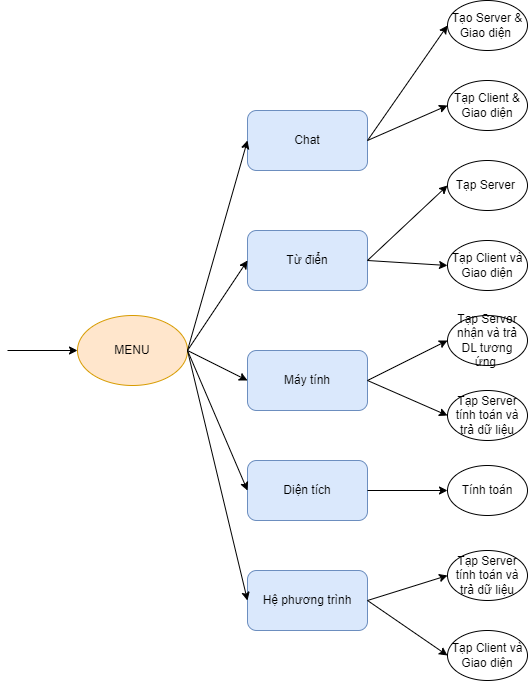
[3.3.3 CHƯƠNG TRÌNH GIẢI HỆ PHƯƠNG TRÌNH 74](#_Toc121601738)

[3.3.4 CHƯƠNG TRÌNH TÍNH TOÁN SỐ HỌC 89](#_Toc121601739)

[TÀI LIỆU THAM KHẢO 99](#_Toc121601740)

CHƯƠNG 1 – PHÂN TÍCH YÊU CẦU

1.1 Tổng quan chương tình



1.2 Giải thích chương trình

Chương trình mô phỏng ứng dụng trao đổi thông tin giữa các máy tính bằng phương thức UDP và TCP.

Cách hoạt động của chương trình:

* Từ menu các user sẽ được cung cấp các sự lựa chọn được thể hiện bằng các Button.
* Với mỗi button khi user click vào sẽ điều hướng đến chương trình mình mong muốn bao gồm:

+ Chat UDP: user sẽ được mô phỏng việc nhắn tin giữa bản thân và server bằng giao thức UDP.

+ Chat TCP: user sẽ được mô phỏng việc nhắn tin giữa bản thân và server bằng giao thức TCP.

+ Từ điển: server sẽ kết nối đến với user qua cổng định sẵn, mỗi khi user nhập từ bằng tiếng Anh sẽ trả về tiếng Việt và ngược lại, chương trình này còn hạn chế vì từ điển chỉ giới hạn trong danh sách tạo sẵn, không lấy dữ liệu trực tiếp.

+ Máy tính: sau khi kết nối với server, user sẽ nhập dữ liệu là phép tính đơn giản, server nhận tính toán và gửi lại kết quả.

+ Diện tích: user sẽ được tính toán diện tích các hình chữ nhật, vuông, bình hành, tam giác, thoi và tròn.

+ Giải hệ phương trình bậc I 2 ẩn x và y: user kết nối đến server, user nhập hệ 2 phương trình gồm các biến a, b, c, server lấy dữ liệu để tính toán sau đó trả lại 2 kết quả là x và y.

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

**- Lập trình mạng là gì?**

• Trong khoa học máy tính, lập trình mạng là giải pháp cho phép áp dụng chương trình hoạt động lên toàn bộ tập các giá trị cùng một lúc. Các giải pháp như vậy thường được dùng trong các thiết lập khoa học và kỹ thuật.

- Internet Addresses Class

Mỗi máy tính kết nối trực tiếp vào Internet phải có địa chỉ IP duy nhất

• Phân biệt địa chỉ public IP và private IP.

+ Ví dụ: 192.618.0.1 là private IP, 81.98.59.133 là public IP.

• Một máy tính có thể có nhiều địa chỉ IP.

• Nếu máy tính nhận địa chỉ 127.0.0.1 thì nó không kết nối với bất kỳ mạng nào đó là địa chỉ local.

**- URLs, URIs Class**

• URI (Uniform Resource Identifier) là một chuỗi chứa các ký tự xác định một tài nguyên vật lý hoặc logic. URI tuân theo các quy tắc cú pháp để đảm bảo tính đồng nhất. Hơn nữa, nó cũng duy trì khả năng mở rộng thông qua một lược đồ đặt tên phân cấp.

• URI bao gồm các phần:

+ Scheme component: Nó là một thành phần trống, theo sau là dấu hai chấm (😃. Scheme chứa một chuỗi các ký tự bắt đầu bằng một chữ cái và theo sau là bất kỳ sự kết hợp nào của chữ số, chữ cái, dấu chấm (.), Dấu gạch ngang (-) hoặc dấu cộng (+). Ví dụ về các scheme nổi tiếng bao gồm HTTP, HTTPS, mailto, tệp, FTP, v.v. Các scheme URI phải được đăng ký với Cơ quan cấp số được ấn định trên Internet (IANA).

+ Authority component: Nó là một trường tùy chọn và đứng trước //.

+ Path: Đường dẫn chứa một chuỗi các phân đoạn được phân tách bằng dấu gạch chéo.

+ Query component: Nó là tùy chọn và đặt trước dấu chấm hỏi (?). Thành phần truy vấn chứa một chuỗi truy vấn dữ liệu không phân cấp.

+ Fragment component: Nó là một trường tùy chọn và được đặt trước bởi một dấu thăng (#). Thành phần phân mảnh bao gồm một mã định danh phân mảnh đưa ra hướng cho tài nguyên thứ cấp.

• URL là một tập hợp con của URI chỉ định nơi tài nguyên tồn tại và cơ chế để truy xuất tài nguyên đó. URL (Uniform Resource Locator) là địa chỉ chung của các documents và giao thức để truy xuất tài nguyên trên mạng máy tính. URL xảy ra thường xuyên nhất khi tham chiếu đến các trang web (HTTP) nhưng cũng có thể được sử dụng để truy cập cơ sở dữ liệu bằng JDBC, email (mailto), truyền tệp (FTP) và nhiều ứng dụng khác.

• URL gồm các thành phần:

+ Protocol: Nó là phần đầu tiên của URL. Ở đây, tên giao thức là Hypertext Transfer Protocol (HTTP).

+ http://www.domainname.com/: Nó là tên miền của bạn. Nó còn được gọi là id máy chủ hoặc host.

+ /folder-name/: Nó chỉ ra rằng trang web được tham chiếu trong một thư mục nhất định trên máy chủ web.

+ web-page-file-name.html: Nó là một tên tệp trang web. ".Html" là extension cho tệp HTML, cho thấy rằng nó là một trang web tĩnh. Tên tệp có thể có các phần mở rộng khác nhau hoặc tùy thuộc vào cách bạn thiết lập máy chủ web. Không thể có phần mở rộng nào cả và URL có thể kết thúc bằng một dòng gạch chéo (/).

• URI là một tập hợp siêu URL xác định tài nguyên bằng URL hoặc URN (Tên tài nguyên thống nhất) hoặc cả hai. Mục đích chính là lấy vị trí hoặc địa chỉ của tài nguyên.

**- Sockets For Clients And Servers**

• Socket là điểm cuối end-point trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server. Các lớp Socket được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.

**- Udp Datagrams And Sockets**

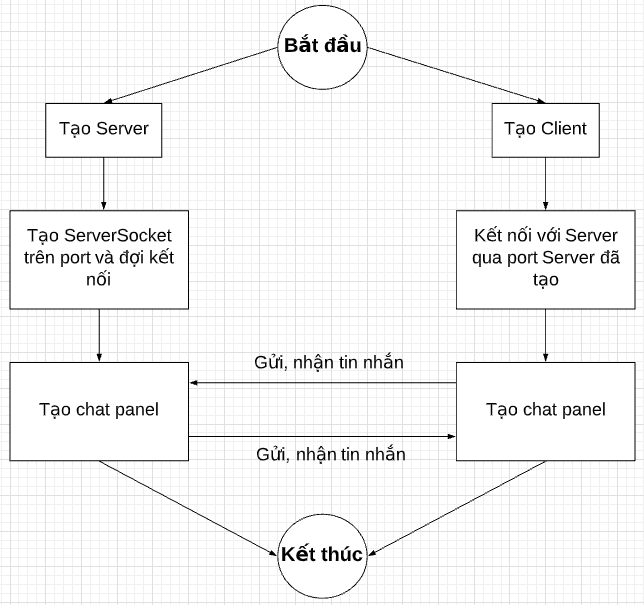
• Datagram Socket hay còn gọi là socket không hướng kết nối, là socket hoạt động thông qua giao thức UDP ( User Datagram Protocol). Datagram Socket có thể hoạt động kể cả khi không có sự thiết lập kết nối giữa 2 máy với nhau. Nói cách khác, đây cũng chính là định nghĩa cho câu hỏi: “Giao thức UDP là gì?“

CHƯƠNG 3 – CHƯƠNG TRÌNH DEMO

TỔNG QUÁT CHƯƠNG TRÌNH (MAIN MENU )

3.1 CHƯƠNG TÌNH CHAT SỬ DỤNG GIAO THỨC TCP

3.1.1 Giới thiệu chương trình



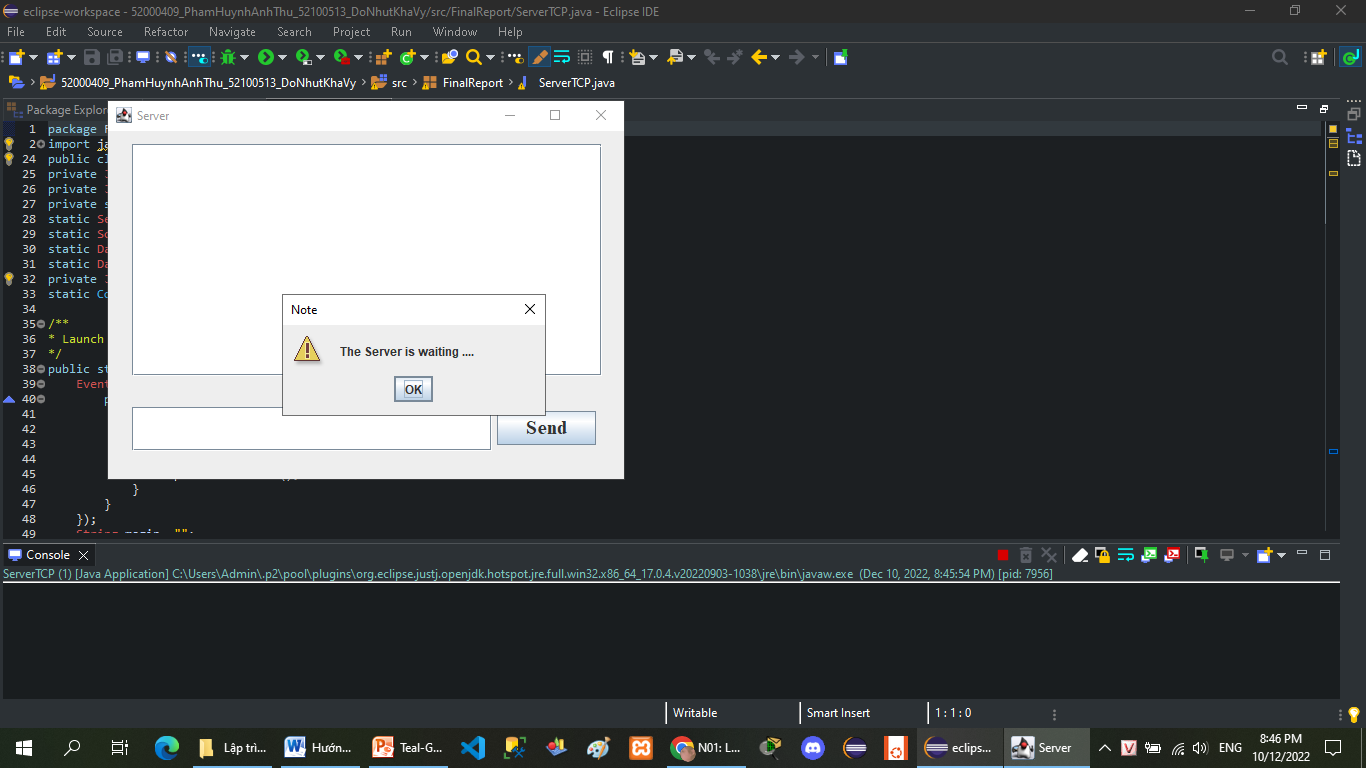
3.1.2 Phân tích biểu đồ

Các bước hoạt động:

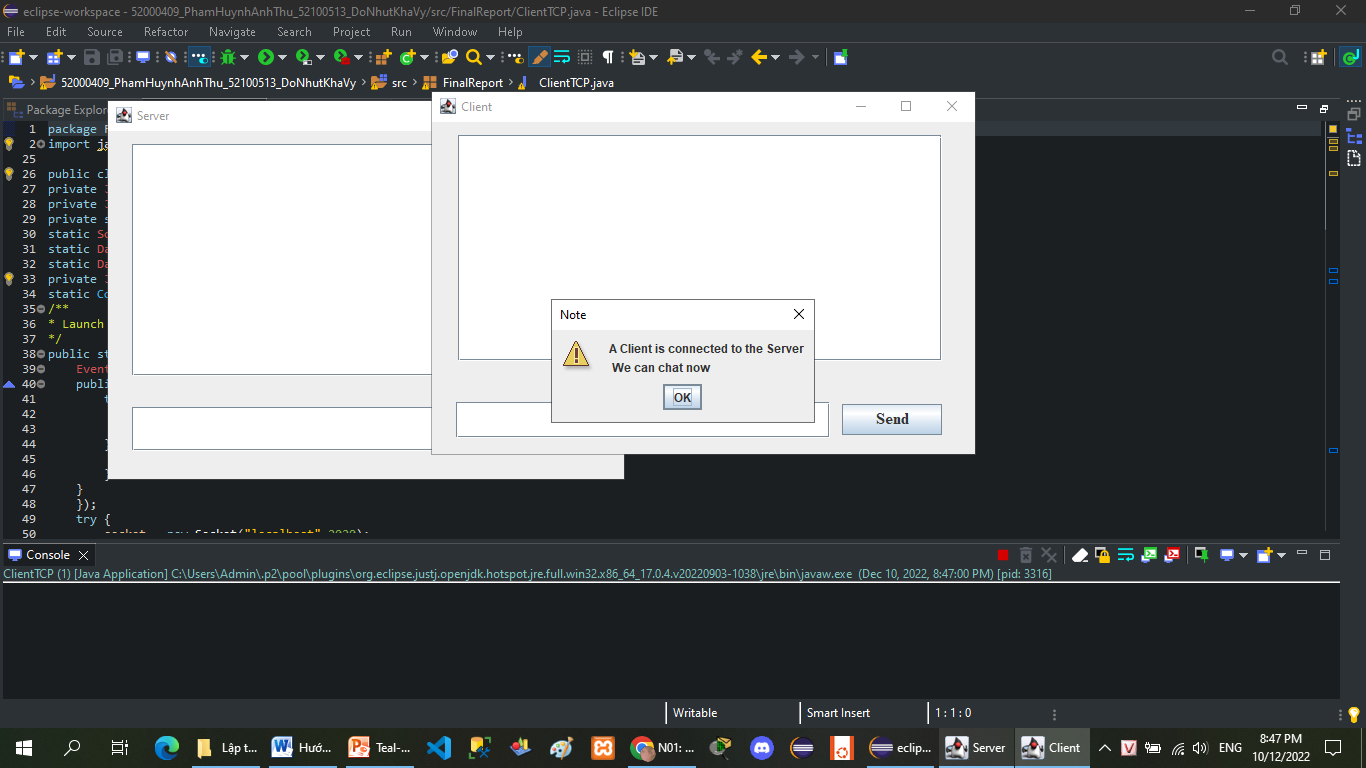
1. Tạo server, client.
2. Tại server tạo một ServerSocket(cổng) – server định sẵn cổng cụ thể và mở để chờ client kết nối. Lúc này server không thể gửi hay nhận tin.
3. Tại client tạo một Socket(hostname, cổng) – hostname của máy client là cục bộ hoặc truyền vào, cổng là cổng của server đang mở.
4. Sau khi client gửi yêu cầu server sẽ chấp nhận kết nối, lúc này server có thể gửi và nhận tin.
5. Sau khi được quyền kết nối đến server client tiến hành gửi tin nhắn và không yêu cầu phải gửi lại tin phản hổi.
6. Server bắt đầu gửi tin cho client hoặc không.
7. Quá trình truyền tin kết thúc, server ngắt mọi kết nối đến cổng.

3.1.3 Giao diện chương trình

**Giao diện Server:**



**Giao diện Client:**



3.1.4 Code chương trình

**Server:**

package FinalReport;

import java.awt.BorderLayout;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JLabel;

import javax.swing.ImageIcon;

import javax.swing.SwingConstants;

public class ServerTCP extends JFrame {

private JPanel contentPane;

private JTextField msg\_text;

private static JTextArea msg\_area;

static ServerSocket serverScoket;

static Socket socket;

static DataInputStream dataIn;

static DataOutputStream dataOut;

private JLabel lblNewLabel;

static Component fame = null;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ServerTCP frame = new ServerTCP();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

String msgin ="";

try {

serverScoket = new ServerSocket(2020);

JOptionPane.showMessageDialog(fame, "The Server is waiting ....",

"Note",JOptionPane.WARNING\_MESSAGE);

socket = serverScoket.accept();

JOptionPane.showMessageDialog(fame, "A Client is connected to the Server\n We can chat now", "Note",JOptionPane.WARNING\_MESSAGE);

dataIn = new DataInputStream(socket.getInputStream());

dataOut = new DataOutputStream(socket.getOutputStream());

while(!msgin.equals("exit")) {

msgin = dataIn.readUTF();

msg\_area.setText(msg\_area.getText().trim()+"\n Client: "+msgin);

}

} catch (IOException e) {

JOptionPane.showMessageDialog(fame, "Sever gap van de ve I/O", "Note",JOptionPane.WARNING\_MESSAGE);

}

finally{

try{

if(socket!=null){

socket.close();

}

}

catch(IOException e){}

}

}

/\*\*

\* Create the frame.

\*/

public ServerTCP() {

setTitle("Server");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 532, 387);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

msg\_area = new JTextArea();

msg\_area.setFont(new Font("Times New Roman", Font.PLAIN, 14));

msg\_area.setEditable(false);

msg\_area.setBounds(24, 13, 470, 232);

contentPane.add(msg\_area);

JScrollPane scrollPane = new JScrollPane(msg\_area);

scrollPane.setBounds(24, 13, 470, 232);

contentPane.add(scrollPane);

msg\_text = new JTextField();

msg\_text.setFont(new Font("Times New Roman", Font.PLAIN, 14));

msg\_text.setBounds(24, 276, 360, 44);

contentPane.add(msg\_text);

msg\_text.setColumns(10);

JButton msg\_send = new JButton("Send");

/\*msg\_send.setHorizontalAlignment(SwingConstants.LEFT);

msg\_send.setIcon(new

ImageIcon(ServerTCP.class.getResource("/com/sun/javafx/webkit/prism/resources/mediaPlayDisabled.png")));\*/

msg\_send.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

String msgOut = " ";

msgOut = msg\_text.getText().trim();

try {

dataOut.writeUTF(msgOut);

msg\_area.setText(msg\_area.getText().trim()+"\n Me: "+msgOut);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

msg\_text.setText("");

}

});

msg\_send.setFont(new Font("Times New Roman", Font.BOLD, 20));

msg\_send.setBounds(389, 280, 99, 34);

contentPane.add(msg\_send);

/\*lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("D:\\Giao An\\Basic NetworkProgramming\\Software\\Hoc Java\\Sever\_Client\_TCP\\Image\\DestTop4.jpg"));

lblNewLabel.setBounds(0, 2, 514, 338);

contentPane.add(lblNewLabel);\*/

}

}

**Client:**

package FinalReport;

import java.awt.BorderLayout;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.net.UnknownHostException;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.ImageIcon;

import javax.swing.SwingConstants;

public class ClientTCP extends JFrame {

private JPanel contentPane;

private JTextField msg\_text;

private static JTextArea textArea\_history;

static Socket socket;

static DataInputStream dataInput;

static DataOutputStream dout;

private JLabel lblNewLabel;

static Component fame = null;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ClientTCP frame = new ClientTCP();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

try {

socket = new Socket("localhost",2020);

dataInput = new DataInputStream(socket.getInputStream());

dout = new DataOutputStream(socket.getOutputStream());

String msgin="";

while(!msgin.equals("exit")) {

msgin = dataInput.readUTF();

textArea\_history.setText(textArea\_history.getText().trim()+"\nServer:"+msgin);

}

} catch (UnknownHostException e) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(fame, "UnknownHostException\n We can chat now", "Note",JOptionPane.WARNING\_MESSAGE);

} catch (IOException e) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(fame, "IOException\n We can not chat now", "Note",JOptionPane.WARNING\_MESSAGE);

}

finally{

try{

if(socket!=null){

socket.close();

}

}

catch(IOException e){}

}

}

/\*\*

\* Create the frame.

\*/

public ClientTCP() {

setTitle("Client");

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 559, 371);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

textArea\_history = new JTextArea();

textArea\_history.setEditable(false);

textArea\_history.setBounds(26, 13, 484, 226);

contentPane.add(textArea\_history);

JScrollPane scrollPane = new JScrollPane(textArea\_history);

scrollPane.setBounds(26, 13, 484, 226);

contentPane.add(scrollPane);

msg\_text = new JTextField();

msg\_text.setBounds(24, 280, 374, 36);

contentPane.add(msg\_text);

msg\_text.setColumns(10);

JButton msg\_send = new JButton("Send");

/\*msg\_send.setHorizontalAlignment(SwingConstants.LEFT);

msg\_send.setIcon(new

ImageIcon(ClientTCP.class.getResource("/com/sun/javafx/webkit/prism/resources/mediaPlayDisabled.png")));\*/

msg\_send.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

String msgOut =" ";

msgOut = msg\_text.getText().trim();

try {

dout.writeUTF(msgOut);

textArea\_history.setText(textArea\_history.getText().trim()+"\n Me: "+msgOut);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

msg\_text.setText("");

}

});

msg\_send.setFont(new Font("Times New Roman", Font.BOLD, 16));

msg\_send.setBounds(410, 282, 100, 31);

contentPane.add(msg\_send);

/\*lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("D:\\Giao An\\Basic NetworkProgramming\\Software\\Hoc Java\\Sever\_Client\_TCP\\Image\\DestTop5.jpg"));

lblNewLabel.setBounds(0, 0, 541, 324);

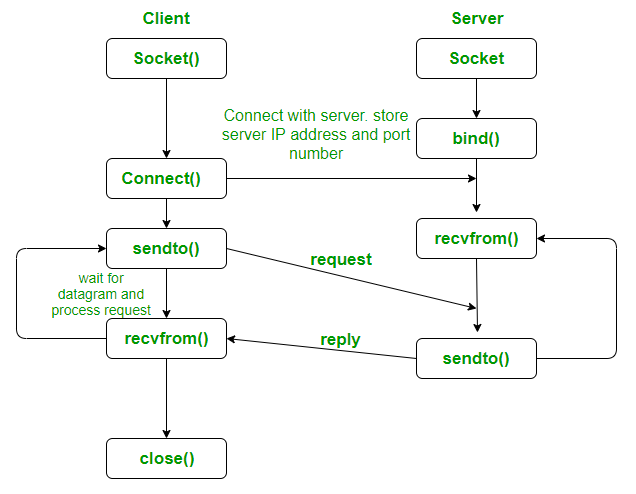
contentPane.add(lblNewLabel);\*/

}

}

3.2 CHƯƠNG TRÌNH CHAT SỬ DỤNG GIAO THỨC UDP

3.1.1 Giới thiệu chương trình



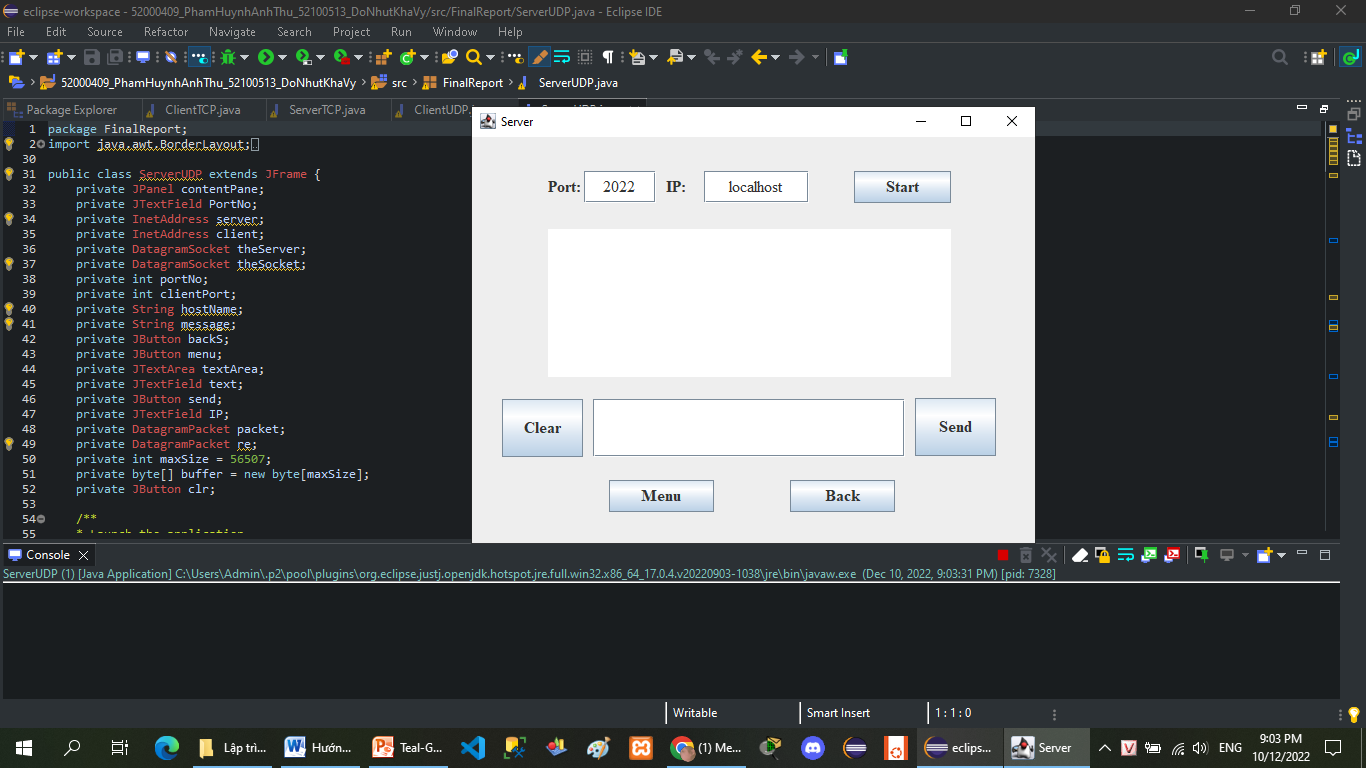
3.1.2 Phân tích biểu đồ

Các bước hoạt động:

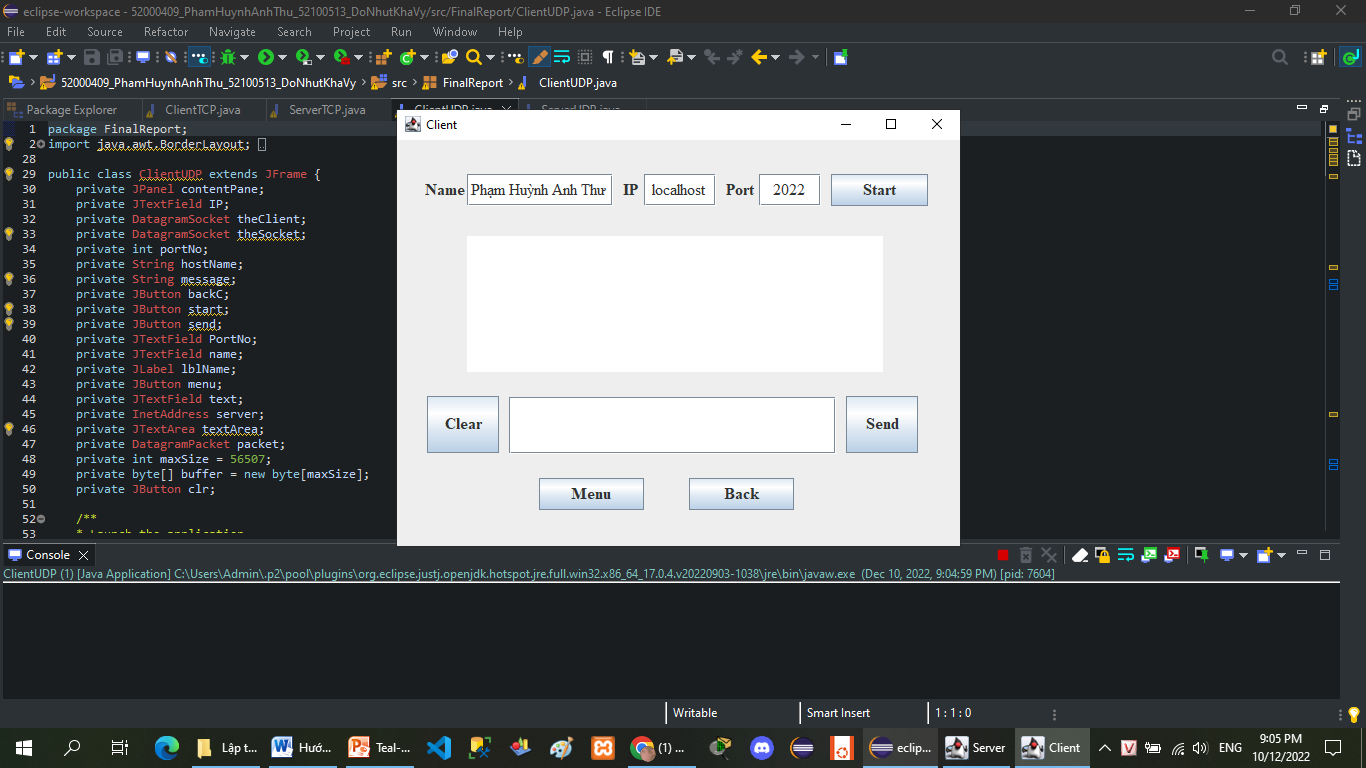
1. Server tạo DatagramSocket với một cổng định sẵn, luôn luôn mở và chấp nhận mọi kết nối.
2. Client tạo một DatagramSocket rỗng, sau khi server mở cổng, client sẽ kết nối đến cổng đó.
3. Client gửi tin nhắn, lúc này client không thể nhận tin nhắn trừ khi server gửi tin phản hồi.
4. Server nhận tin nhắn và gửi lại tin phản hổi cho client.
5. Client nhận được tin nhắn, client có thể tiếp tục gửi tin nhắn và đợi phản hổi hoặc kết thúc ngay khi vừa nhận được tin
6. Client ngắt kết nối.

3.1.3 Giao diện chương trình

**Giao diện Server:**



**Giao diện Client:**



3.1.4 Code chương trình

**Server:**

package FinalReport;

import java.awt.BorderLayout;

import java.net.\*;

import java.io.\*;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.awt.event.ActionEvent;

import javax.swing.JTextArea;

import javax.swing.SwingConstants;

import javax.swing.JTabbedPane;

import javax.swing.event.AncestorListener;

import javax.swing.event.AncestorEvent;

public class ServerUDP extends JFrame {

private JPanel contentPane;

private JTextField PortNo;

private InetAddress server;

private InetAddress client;

private DatagramSocket theServer;

private DatagramSocket theSocket;

private int portNo;

private int clientPort;

private String hostName;

private String message;

private JButton backS;

private JButton menu;

private JTextArea textArea;

private JTextField text;

private JButton send;

private JTextField IP;

private DatagramPacket packet;

private DatagramPacket re;

private int maxSize = 56507;

private byte[] buffer = new byte[maxSize];

private JButton clr;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ServerUDP frame = new ServerUDP();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\* @throws IOException

\*/

public void rec(DatagramPacket p) {

//get message from client for address

try {

theServer.receive(p);

//rec message and display

String message ="";

Component fame = null;

message = new String(p.getData(), 0, p.getLength( ), "UTF-8");

textArea.setText(textArea.getText().trim()+"\n"

+p.getAddress( ) + " at port "

+ p.getPort( ) +" say: " + message);

JOptionPane.showMessageDialog(fame, "New data is received");

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public ServerUDP() {

setTitle("Server");

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 579, 445);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

PortNo = new JTextField();

PortNo.setHorizontalAlignment(SwingConstants.CENTER);

PortNo.setText("2022");

PortNo.setFont(new Font("Times New Roman", Font.PLAIN, 16));

PortNo.setBounds(112, 34, 72, 32);

contentPane.add(PortNo);

PortNo.setColumns(10);

JLabel lblNewLabel = new JLabel("Port:");

lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblNewLabel.setBounds(76, 34, 45, 32);

contentPane.add(lblNewLabel);

JButton start = new JButton("Start");

start.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//

//

portNo = Integer.parseInt(PortNo.getText());

String getData = text.getText();

byte[] data = getData.getBytes();

try {

textArea.setText(textArea.getText().trim()+"\n" + "Starting chat...");

theServer = new DatagramSocket(portNo);

Component fame = null;

JOptionPane.showMessageDialog(fame, "Ready to receive data"," UDP Protocal",JOptionPane.WARNING\_MESSAGE);

//get message from client for address

packet = new DatagramPacket(buffer, buffer.length);

rec(packet);

//send conncet accept

} catch (SocketException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

theServer.close();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

theServer.close();

}

text.setText("");

}

});

start.setFont(new Font("Times New Roman", Font.BOLD, 16));

start.setBounds(382, 34, 97, 32);

contentPane.add(start);

backS = new JButton("Back");

backS.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ServerUDP frame = new ServerUDP();

frame.setVisible(true);

}

});

backS.setFont(new Font("Times New Roman", Font.BOLD, 16));

backS.setBounds(318, 343, 105, 32);

contentPane.add(backS);

menu = new JButton("Menu");

menu.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

Menu frame = new Menu();

frame.setVisible(true);

}

});

menu.setFont(new Font("Times New Roman", Font.BOLD, 16));

menu.setBounds(137, 343, 105, 32);

contentPane.add(menu);

textArea = new JTextArea();

textArea.setBounds(76, 92, 403, 148);

contentPane.add(textArea);

text = new JTextField();

text.setText("");

text.setBounds(121, 262, 312, 58);

contentPane.add(text);

text.setColumns(10);

send = new JButton("Send");

send.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

portNo = Integer.parseInt(PortNo.getText());

try {

Component fame = null;

//send

String getdata = text.getText();

client = packet.getAddress();

clientPort = packet.getPort();

theServer = new DatagramSocket();

byte[] dataS = getdata.getBytes();

DatagramPacket packetS = new DatagramPacket(dataS, dataS.length, client, clientPort);

theServer.send(packetS);

textArea.setText(textArea.getText().trim() + "\n" + "Server say: " + getdata);

text.setText("");

}

catch (SocketException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

text.setText("");

}

});

send.setFont(new Font("Times New Roman", Font.BOLD, 16));

send.setBounds(443, 261, 81, 58);

contentPane.add(send);

JLabel lblIp = new JLabel("IP:");

lblIp.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblIp.setBounds(194, 34, 45, 32);

contentPane.add(lblIp);

IP = new JTextField();

IP.setText("localhost");

IP.setHorizontalAlignment(SwingConstants.CENTER);

IP.setFont(new Font("Times New Roman", Font.PLAIN, 16));

IP.setColumns(10);

IP.setBounds(232, 34, 105, 32);

contentPane.add(IP);

clr = new JButton("Clear");

clr.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textArea.setText("");

}

});

clr.setFont(new Font("Times New Roman", Font.BOLD, 16));

clr.setBounds(30, 262, 81, 58);

contentPane.add(clr);

}

}

**Client:**

package FinalReport;

import java.awt.BorderLayout;

import java.net.\*;

import java.io.\*;

import java.lang.\*;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.awt.event.ActionEvent;

import javax.swing.JTextArea;

import javax.swing.SwingConstants;

public class ClientUDP extends JFrame {

private JPanel contentPane;

private JTextField IP;

private DatagramSocket theClient;

private DatagramSocket theSocket;

private int portNo;

private String hostName;

private String message;

private JButton backC;

private JButton start;

private JButton send;

private JTextField PortNo;

private JTextField name;

private JLabel lblName;

private JButton menu;

private JTextField text;

private InetAddress server;

private JTextArea textArea;

private DatagramPacket packet;

private int maxSize = 56507;

private byte[] buffer = new byte[maxSize];

private JButton clr;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ClientUDP frame = new ClientUDP();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public ClientUDP() {

setTitle("Client");

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 579, 445);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

IP = new JTextField();

IP.setHorizontalAlignment(SwingConstants.CENTER);

IP.setText("localhost");

IP.setFont(new Font("Times New Roman", Font.PLAIN, 16));

IP.setBounds(247, 34, 72, 32);

contentPane.add(IP);

IP.setColumns(10);

JLabel lblNewLabel = new JLabel("Port");

lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblNewLabel.setBounds(329, 34, 34, 32);

contentPane.add(lblNewLabel);

JButton start = new JButton("Start");

start.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

hostName = IP.getText();

portNo = Integer.parseInt(PortNo.getText());

String getdata = text.getText();

try {

//connect

server = InetAddress.getByName(hostName);

theClient = new DatagramSocket();

Component fame = null;

JOptionPane.showMessageDialog(fame, "Ready to send data","UDP Protocal",JOptionPane.WARNING\_MESSAGE);

} catch (SocketException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

start.setFont(new Font("Times New Roman", Font.BOLD, 16));

start.setBounds(434, 34, 97, 32);

contentPane.add(start);

backC = new JButton("Back");

backC.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ClientUDP frame = new ClientUDP();

frame.setVisible(true);

}

});

backC.setFont(new Font("Times New Roman", Font.BOLD, 16));

backC.setBounds(292, 338, 105, 32);

contentPane.add(backC);

PortNo = new JTextField();

PortNo.setText("2022");

PortNo.setHorizontalAlignment(SwingConstants.CENTER);

PortNo.setFont(new Font("Times New Roman", Font.PLAIN, 16));

PortNo.setColumns(10);

PortNo.setBounds(362, 34, 62, 32);

contentPane.add(PortNo);

JLabel lblIp = new JLabel("IP");

lblIp.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblIp.setBounds(226, 34, 25, 32);

contentPane.add(lblIp);

name = new JTextField();

name.setText("Phạm Huỳnh Anh Thư");

name.setHorizontalAlignment(SwingConstants.CENTER);

name.setFont(new Font("Times New Roman", Font.PLAIN, 16));

name.setColumns(10);

name.setBounds(70, 34, 146, 32);

contentPane.add(name);

lblName = new JLabel("Name");

lblName.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblName.setBounds(28, 34, 54, 32);

contentPane.add(lblName);

menu = new JButton("Menu");

menu.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Menu frame = new Menu();

frame.setVisible(true);

}

});

menu.setFont(new Font("Times New Roman", Font.BOLD, 16));

menu.setBounds(142, 338, 105, 32);

contentPane.add(menu);

JTextArea textArea = new JTextArea();

textArea.setBounds(70, 96, 416, 136);

contentPane.add(textArea);

text = new JTextField();

text.setBounds(112, 257, 327, 57);

contentPane.add(text);

text.setColumns(10);

JButton send = new JButton("Send");

send.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getdata = text.getText();

Component fame = null;

try {

packet = new DatagramPacket(buffer, buffer.length);

byte[] data = getdata.getBytes();

//send

getdata = text.getText();

data = getdata.getBytes();

DatagramPacket packetS = new DatagramPacket(data, data.length, server, portNo);

theClient.send(packetS);

textArea.setText(textArea.getText().trim()+"\n Me say: " + getdata);

text.setText("");

theClient.receive(packet);

String rep = new String(packet.getData(), 0, packet.getLength());

textArea.setText(textArea.getText().trim()+"\n" + "|\_\_Server says: " + rep);

packetS = new DatagramPacket(data, data.length, server, portNo);

theClient.send(packetS);

/\*if(getdata.equals("Exit")) {

JOptionPane.showMessageDialog(fame, "End chat!");

theClient.send(packetS);

theClient.close();

}\*/

} catch (UnsupportedEncodingException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

text.setText("");

}

});

send.setFont(new Font("Times New Roman", Font.BOLD, 16));

send.setBounds(449, 256, 72, 57);

contentPane.add(send);

clr = new JButton("Clear");

clr.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textArea.setText("");

}

});

clr.setFont(new Font("Times New Roman", Font.BOLD, 16));

clr.setBounds(30, 256, 72, 57);

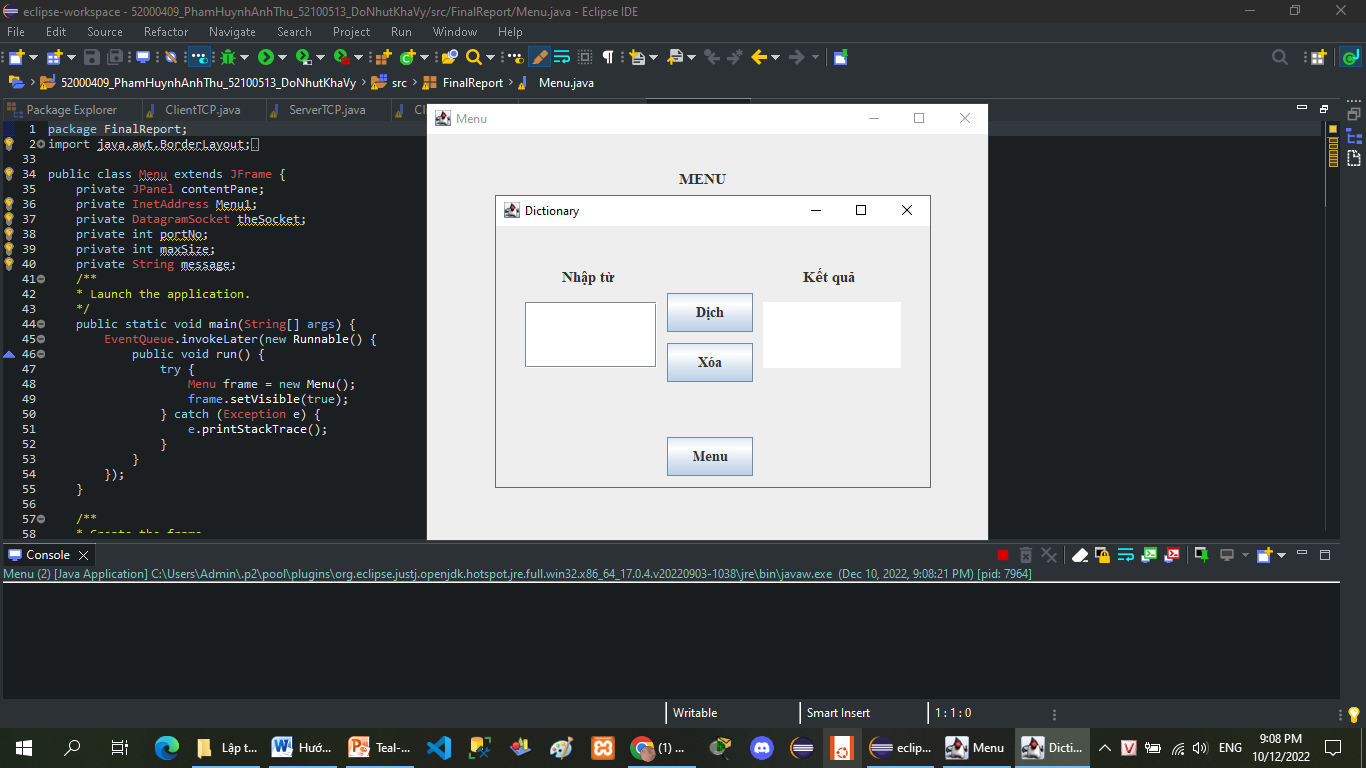
contentPane.add(clr);

}

}

3.3 CÁC CHƯƠNG TRINHG ỨNG DỤNG KHÁC

3.3.1 CHƯƠNG TÌNH TỪ ĐIỂN



**Giải thích chương trình:**

1. Tạo một server Socket với cổng 2022 và chờ kết nối.
2. Tạo client Socket(hostname, cổng).
3. Server chấp nhận kết nối.
4. Từ giao diện của client nhập từ cần tra, DataInputStream() nhận dữ liệu.
5. Từ dữ liệu của client server sẽ thực hiện tìm nghĩa của từ đó có trong HashMap, nếu có ghi ra file DataOutputStream() và gửi lại client.
6. Client có thể tra tiếp hoặc ngừng kết nối.

**Code chương trình:**

***Giao diện client:***

package FinalReport;

import java.awt.BorderLayout;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.PrintWriter;

import java.io.UnsupportedEncodingException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.Socket;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.util.\*;

import java.awt.event.ActionEvent;

import javax.swing.JTextArea;

import javax.swing.SwingConstants;

public class Dictionary extends JFrame{

private JTextField text;

private JTextArea res;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Dictionary frame = new Dictionary();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\*

\*/

public Dictionary() {

setTitle("Dictionary");

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("Nhập từ");

lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 15));

lblNewLabel.setBounds(66, 37, 69, 28);

getContentPane().add(lblNewLabel);

text = new JTextField();

text.setHorizontalAlignment(SwingConstants.CENTER);

text.setBounds(29, 76, 132, 66);

getContentPane().add(text);

text.setColumns(10);

JLabel lblKtQu = new JLabel("Kết quả");

lblKtQu.setFont(new Font("Times New Roman", Font.BOLD, 15));

lblKtQu.setBounds(307, 37, 60, 28);

getContentPane().add(lblKtQu);

JButton trans = new JButton("Dịch");

trans.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

//

String word = text.getText();

String result = "";

if(word.length() == 0){

result += "";

res.setText(result);

JOptionPane.showMessageDialog(null, "Vui lòng nhập từ hoặc từ không có trong từ điển.");

}

//Neu khong rong

else {

//connect

Socket dic = new Socket("localhost", 2022);

//tao inputstream chua ket qua tu server

Scanner in = new Scanner(dic.getInputStream());

//tao outputstream chua word gui den server

PrintWriter out = new PrintWriter(dic.getOutputStream(),true);

out.println(word);

//lay result tu in

result = in.nextLine();

res.setText(result);

}

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

trans.setFont(new Font("Times New Roman", Font.BOLD, 14));

trans.setBounds(171, 67, 86, 39);

getContentPane().add(trans);

res = new JTextArea();

res.setColumns(2);

res.setFont(new Font("Times New Roman", Font.PLAIN, 14));

res.setBounds(267, 76, 138, 66);

getContentPane().add(res);

JButton back = new JButton("Menu");

back.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

Menu frame = new Menu();

frame.setVisible(true);

}

});

back.setFont(new Font("Times New Roman", Font.BOLD, 14));

back.setBounds(171, 211, 86, 39);

getContentPane().add(back);

JButton del = new JButton("Xóa");

del.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String word = text.getText();

String result = "";

text.setText("");

res.setText("");

}

});

del.setFont(new Font("Times New Roman", Font.BOLD, 14));

del.setBounds(171, 117, 86, 39);

getContentPane().add(del);

}

}

***Server:***

package FinalReport;

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class DicServer {

public static void main(String[] args) {

try{

final ServerSocket svr = new ServerSocket(2022);

System.out.println("Server listening on port 2022...");

while(true){

Socket d = svr.accept();

new Thread(new DicRunnable(d)).start();

}

}catch(Exception ex){

ex.printStackTrace();

}

}

}

class DicRunnable implements Runnable{

Socket client = null;

DicRunnable(Socket client){

this.client = client;

}

@Override

public void run() {

try {

//lay word tu client

InputStream is = client.getInputStream();

Scanner in = new Scanner(is);

//tra ket qua

OutputStream os = client.getOutputStream();

PrintWriter out = new PrintWriter(os,true/\*auto-flush\*/);

while(in.hasNextLine()){

String word = in.nextLine();

String result = returnValue(word);

out.println(result);

}

} catch (Exception e) {

e.printStackTrace();

}

}

private String returnValue(String s) {

Map<String, String> dic = new HashMap<String, String>();

dic.put("con chó", "dog");

dic.put("con mèo", "cat");

dic.put("con gà", "chicken");

dic.put("con vịt", "duck");

dic.put("con heo", "pig");

dic.put("con cáo", "fox");

dic.put("con gấu", "bear");

dic.put("con voi", "elephant");

dic.put("con sư tử", "lion");

dic.put("con nhím", "porcupine");

String result = "";

//Khoi tao ArrayList chua key

List<String> keyL = new ArrayList<String>(dic.keySet());

//Khoi tao ArrayList chua value

List<String> valueL = new ArrayList<String>();

for(String key1 : keyL) {

valueL.add(dic.get(key1));

}

try {

//Check neu input rong thi yeu cau nhap lai

//Check neu list key chua word thi lay ra value tuong ung tu key

for(String key : keyL) {

if(key.contains(s)){

result += dic.get(key);

}

}

//Hoac lay ra key tuong ung tu value

for(String value :valueL) {

if(value.contains(s)){

result += keyL.get(valueL.indexOf(value));

}

}

} catch (Exception e1) {

e1.printStackTrace();

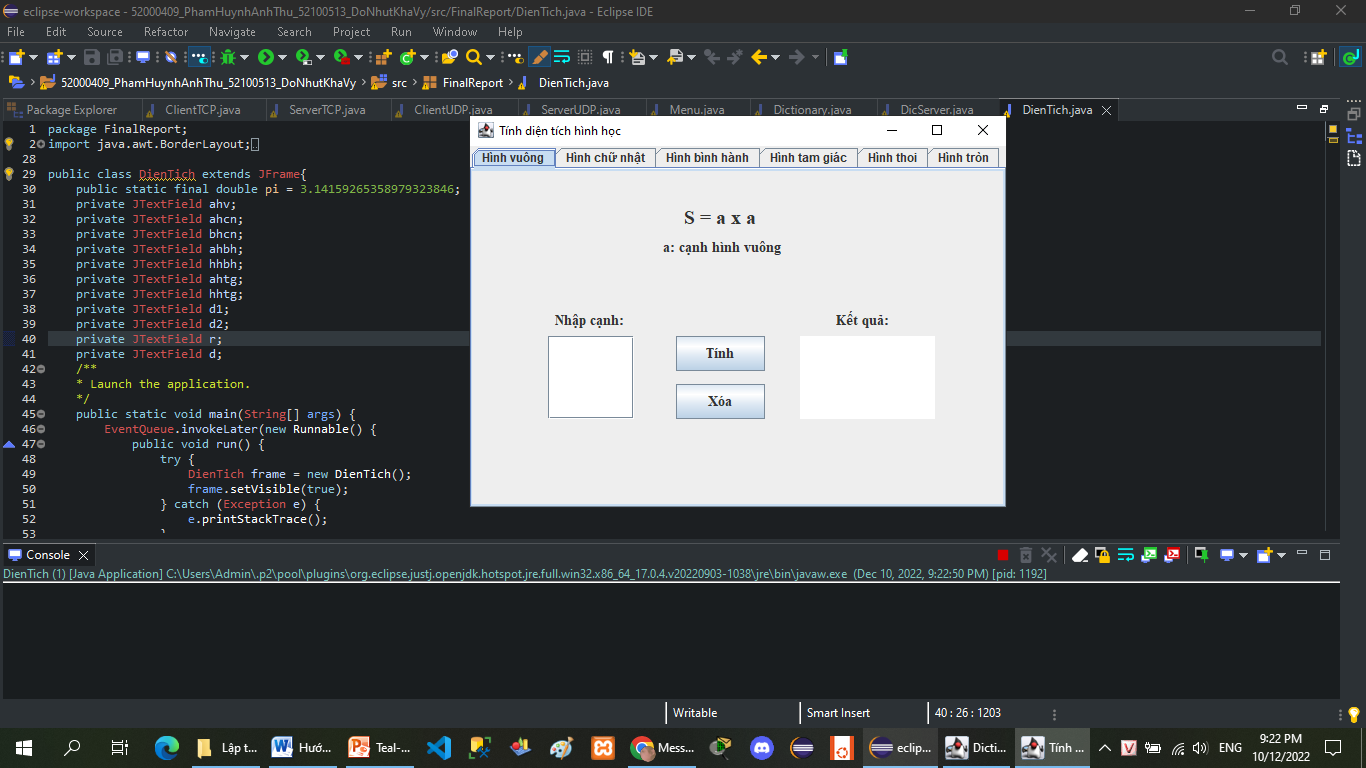
}

return result;

}

}

3.3.2 CHƯƠNG TRÌNH TÍNH TIỆN TÍCH HÌNH HOC



**Code chương trình:**

package FinalReport;

import java.awt.BorderLayout;

import java.awt.\*;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.util.\*;

import java.awt.event.ActionEvent;

import javax.swing.JTextArea;

import javax.swing.SwingConstants;

import javax.swing.JTabbedPane;

public class DienTich extends JFrame{

public static final double pi = 3.14159265358979323846;

private JTextField ahv;

private JTextField ahcn;

private JTextField bhcn;

private JTextField ahbh;

private JTextField hhbh;

private JTextField ahtg;

private JTextField hhtg;

private JTextField d1;

private JTextField d2;

private JTextField r;

private JTextField d;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

DienTich frame = new DienTich();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\*

\*/

/\*public void checkInput() {

if(ahv.getText().isEmpty() ||

ahcn.getText().isEmpty() || bhcn.getText().isEmpty() ||

ahbh.getText().isEmpty() || hhbh.getText().isEmpty() ||

ahtg.getText().isEmpty() || hhtg.getText().isEmpty() ||

d1.getText().isEmpty() || d2.getText().isEmpty() ||

r.getText().isEmpty()) {

//Show message

Component fame = null;

JOptionPane.showMessageDialog(fame, "Vui lòng không nhập trống.", null, JOptionPane.WARNING\_MESSAGE);

}

}\*/

public DienTich() {

setTitle("Tính diện tích hình học");

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 552, 400);

JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);

getContentPane().add(tabbedPane, BorderLayout.CENTER);

//Hinh vuong

JPanel hv = new JPanel();

tabbedPane.addTab("Hình vuông", null, hv, null);

hv.setLayout(null);

JLabel lblNewLabel = new JLabel("S = a x a ");

lblNewLabel.setBounds(178, 23, 145, 48);

lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 20));

hv.add(lblNewLabel);

JLabel lblACnhHnh = new JLabel("a: cạnh hình vuông");

lblACnhHnh.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh.setBounds(178, 52, 145, 48);

hv.add(lblACnhHnh);

JLabel lblNhpCnh = new JLabel("Nhập cạnh:");

lblNhpCnh.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh.setBounds(45, 125, 145, 48);

hv.add(lblNhpCnh);

ahv = new JTextField();

ahv.setFont(new Font("Times New Roman", Font.PLAIN, 14));

ahv.setHorizontalAlignment(SwingConstants.CENTER);

ahv.setBounds(76, 165, 86, 83);

hv.add(ahv);

ahv.setColumns(10);

JTextArea Shv = new JTextArea();

Shv.setBounds(328, 165, 135, 83);

hv.add(Shv);

JButton calhv = new JButton("Tính");

calhv.setFont(new Font("Times New Roman", Font.BOLD, 14));

calhv.setBounds(204, 165, 89, 35);

hv.add(calhv);

calhv.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//checkInput();

double aHV = Double.valueOf(ahv.getText());

double SHV = 0.0;

SHV = aHV \* aHV;

Shv.setText(Double.toString(SHV));

}

});

JLabel lblKtQu = new JLabel("Kết quả:");

lblKtQu.setHorizontalAlignment(SwingConstants.CENTER);

lblKtQu.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblKtQu.setBounds(318, 125, 145, 48);

hv.add(lblKtQu);

JButton delhv = new JButton("Xóa");

delhv.setFont(new Font("Times New Roman", Font.BOLD, 14));

delhv.setBounds(204, 213, 89, 35);

hv.add(delhv);

delhv.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ahv.setText("");

Shv.setText("");

}

});

//Hinh chu nhat

JPanel hcn = new JPanel();

tabbedPane.addTab("Hình chữ nhật", null, hcn, null);

hcn.setLayout(null);

JLabel lblNewLabel1 = new JLabel("S = a x b ");

lblNewLabel1.setBounds(178, 23, 145, 48);

lblNewLabel1.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel1.setFont(new Font("Times New Roman", Font.BOLD, 20));

hcn.add(lblNewLabel1);

JLabel lblACnhHnh1a = new JLabel("a: chiều dài");

lblACnhHnh1a.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh1a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh1a.setBounds(178, 52, 145, 48);

hcn.add(lblACnhHnh1a);

JLabel lblACnhHnh1b = new JLabel("b: chiều rộng");

lblACnhHnh1b.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh1b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh1b.setBounds(178, 82, 145, 48);

hcn.add(lblACnhHnh1b);

JLabel lblNhpCnh1a = new JLabel("Nhập chiều dài:");

lblNhpCnh1a.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh1a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh1a.setBounds(49, 125, 145, 48);

hcn.add(lblNhpCnh1a);

JLabel lblNhpCnh1b = new JLabel("Nhập chiều rộng:");

lblNhpCnh1b.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh1b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh1b.setBounds(49, 200, 145, 48);

hcn.add(lblNhpCnh1b);

ahcn = new JTextField();

ahcn.setFont(new Font("Times New Roman", Font.PLAIN, 14));

ahcn.setHorizontalAlignment(SwingConstants.CENTER);

ahcn.setBounds(79, 165, 86, 35);

hcn.add(ahcn);

ahcn.setColumns(10);

JTextArea Shcn = new JTextArea();

Shcn.setBounds(328, 165, 135, 111);

hcn.add(Shcn);

JButton calhcn = new JButton("Tính");

calhcn.setFont(new Font("Times New Roman", Font.BOLD, 14));

calhcn.setBounds(204, 180, 89, 35);

hcn.add(calhcn);

calhcn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//checkInput();

double aHCN = Double.valueOf(ahcn.getText());

double bHCN = Double.valueOf(bhcn.getText());

double SHCN = 0.0;

SHCN = aHCN \* bHCN;

Shcn.setText(Double.toString(SHCN));

}

});

JLabel lblKtQu1 = new JLabel("Kết quả:");

lblKtQu1.setHorizontalAlignment(SwingConstants.CENTER);

lblKtQu1.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblKtQu1.setBounds(318, 125, 145, 48);

hcn.add(lblKtQu1);

JButton delhcn = new JButton("Xóa");

delhcn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ahcn.setText("");

bhcn.setText("");

Shcn.setText("");

}

});

delhcn.setFont(new Font("Times New Roman", Font.BOLD, 14));

delhcn.setBounds(204, 229, 89, 35);

hcn.add(delhcn);

bhcn = new JTextField();

bhcn.setHorizontalAlignment(SwingConstants.CENTER);

bhcn.setFont(new Font("Times New Roman", Font.PLAIN, 14));

bhcn.setColumns(10);

bhcn.setBounds(79, 241, 86, 35);

hcn.add(bhcn);

//Hinh binh hanh

JPanel hbh = new JPanel();

tabbedPane.addTab("Hình bình hành", null, hbh, null);

hbh.setLayout(null);

JLabel lblNewLabel2 = new JLabel("S = a x h ");

lblNewLabel2.setBounds(178, 23, 145, 48);

lblNewLabel2.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel2.setFont(new Font("Times New Roman", Font.BOLD, 20));

hbh.add(lblNewLabel2);

JLabel lblACnhHnh2a = new JLabel("a: cạnh đáy");

lblACnhHnh2a.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh2a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh2a.setBounds(178, 52, 145, 48);

hbh.add(lblACnhHnh2a);

JLabel lblACnhHnh2b = new JLabel("h: đường cao");

lblACnhHnh2b.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh2b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh2b.setBounds(178, 82, 145, 48);

hbh.add(lblACnhHnh2b);

JLabel lblNhpCnh2a = new JLabel("Nhập cạnh đáy:");

lblNhpCnh2a.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh2a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh2a.setBounds(49, 125, 145, 48);

hbh.add(lblNhpCnh2a);

JLabel lblNhpCnh2b = new JLabel("Nhập đường cao:");

lblNhpCnh2b.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh2b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh2b.setBounds(49, 200, 145, 48);

hbh.add(lblNhpCnh2b);

ahbh = new JTextField();

ahbh.setFont(new Font("Times New Roman", Font.PLAIN, 14));

ahbh.setHorizontalAlignment(SwingConstants.CENTER);

ahbh.setBounds(79, 165, 86, 35);

hbh.add(ahbh);

ahbh.setColumns(10);

JTextArea Shbh = new JTextArea();

Shbh.setBounds(328, 165, 135, 111);

hbh.add(Shbh);

hhbh = new JTextField();

hhbh.setHorizontalAlignment(SwingConstants.CENTER);

hhbh.setFont(new Font("Times New Roman", Font.PLAIN, 14));

hhbh.setColumns(10);

hhbh.setBounds(79, 241, 86, 35);

hbh.add(hhbh);

JButton calhbh = new JButton("Tính");

calhbh.setFont(new Font("Times New Roman", Font.BOLD, 14));

calhbh.setBounds(204, 180, 89, 35);

hbh.add(calhbh);

calhbh.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//checkInput();

double aHBH = Double.valueOf(ahbh.getText());

double hHBH = Double.valueOf(hhbh.getText());

double SHBH = 0.0;

SHBH = aHBH \* hHBH;

Shbh.setText(Double.toString(SHBH));

}

});

JLabel lblKtQu2 = new JLabel("Kết quả:");

lblKtQu2.setHorizontalAlignment(SwingConstants.CENTER);

lblKtQu2.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblKtQu2.setBounds(318, 125, 145, 48);

hbh.add(lblKtQu2);

JButton delhbh = new JButton("Xóa");

delhbh.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ahbh.setText("");

hhbh.setText("");

Shbh.setText("");

}

});

delhbh.setFont(new Font("Times New Roman", Font.BOLD, 14));

delhbh.setBounds(204, 229, 89, 35);

hbh.add(delhbh);

//Hinh tam giac

JPanel htg = new JPanel();

tabbedPane.addTab("Hình tam giác", null, htg, null);

htg.setLayout(null);

JLabel lblNewLabel3 = new JLabel("S = (a x h) / 2 ");

lblNewLabel3.setBounds(178, 23, 145, 48);

lblNewLabel3.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel3.setFont(new Font("Times New Roman", Font.BOLD, 20));

htg.add(lblNewLabel3);

JLabel lblACnhHnh3a = new JLabel("a: cạnh đáy");

lblACnhHnh3a.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh3a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh3a.setBounds(178, 52, 145, 48);

htg.add(lblACnhHnh3a);

JLabel lblACnhHnh3b = new JLabel("h: đường cao");

lblACnhHnh3b.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh3b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh3b.setBounds(178, 82, 145, 48);

htg.add(lblACnhHnh3b);

JLabel lblNhpCnh3a = new JLabel("Nhập cạnh đáy:");

lblNhpCnh3a.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh3a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh3a.setBounds(49, 125, 145, 48);

htg.add(lblNhpCnh3a);

JLabel lblNhpCnh3b = new JLabel("Nhập đường cao:");

lblNhpCnh3b.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh3b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh3b.setBounds(49, 200, 145, 48);

htg.add(lblNhpCnh3b);

ahtg = new JTextField();

ahtg.setFont(new Font("Times New Roman", Font.PLAIN, 14));

ahtg.setHorizontalAlignment(SwingConstants.CENTER);

ahtg.setBounds(79, 165, 86, 35);

htg.add(ahtg);

ahtg.setColumns(10);

JTextArea Shtg = new JTextArea();

Shtg.setBounds(328, 165, 135, 111);

htg.add(Shtg);

hhtg = new JTextField();

hhtg.setHorizontalAlignment(SwingConstants.CENTER);

hhtg.setFont(new Font("Times New Roman", Font.PLAIN, 14));

hhtg.setColumns(10);

hhtg.setBounds(79, 241, 86, 35);

htg.add(hhtg);

JButton calhtg = new JButton("Tính");

calhtg.setFont(new Font("Times New Roman", Font.BOLD, 14));

calhtg.setBounds(204, 180, 89, 35);

htg.add(calhtg);

calhtg.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//checkInput();

double aHTG = Double.valueOf(ahtg.getText());

double hHTG = Double.valueOf(hhtg.getText());

double SHTG = 0.0;

SHTG = (aHTG \* hHTG) / 2;

Shtg.setText(Double.toString(SHTG));

}

});

JLabel lblKtQu3 = new JLabel("Kết quả:");

lblKtQu3.setHorizontalAlignment(SwingConstants.CENTER);

lblKtQu3.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblKtQu3.setBounds(318, 125, 145, 48);

htg.add(lblKtQu3);

JButton delhtg = new JButton("Xóa");

delhtg.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ahtg.setText("");

hhtg.setText("");

Shtg.setText("");

}

});

delhtg.setFont(new Font("Times New Roman", Font.BOLD, 14));

delhtg.setBounds(204, 229, 89, 35);

htg.add(delhtg);

//Hinh thoi

JPanel hthoi = new JPanel();

tabbedPane.addTab("Hình thoi", null, hthoi, null);

hthoi.setLayout(null);

JLabel lblNewLabel4 = new JLabel("S = (d1 x d2) / 2 ");

lblNewLabel4.setBounds(178, 23, 145, 48);

lblNewLabel4.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel4.setFont(new Font("Times New Roman", Font.BOLD, 20));

hthoi.add(lblNewLabel4);

JLabel lblACnhHnh4a = new JLabel("d1: đường chéo thứ nhất");

lblACnhHnh4a.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh4a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh4a.setBounds(154, 52, 193, 48);

hthoi.add(lblACnhHnh4a);

JLabel lblACnhHnh4b = new JLabel("d2: đường chéo thứ hai");

lblACnhHnh4b.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh4b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh4b.setBounds(154, 80, 183, 48);

hthoi.add(lblACnhHnh4b);

JLabel lblNhpCnh4a = new JLabel("Nhập đường chéo I:");

lblNhpCnh4a.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh4a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh4a.setBounds(49, 125, 145, 48);

hthoi.add(lblNhpCnh4a);

JLabel lblNhpCnh4b = new JLabel("Nhập đường chéo II:");

lblNhpCnh4b.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh4b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh4b.setBounds(49, 200, 145, 48);

hthoi.add(lblNhpCnh4b);

d1 = new JTextField();

d1.setFont(new Font("Times New Roman", Font.PLAIN, 14));

d1.setHorizontalAlignment(SwingConstants.CENTER);

d1.setBounds(79, 165, 86, 35);

hthoi.add(d1);

d1.setColumns(10);

JTextArea Shthoi = new JTextArea();

Shthoi.setBounds(328, 165, 135, 111);

hthoi.add(Shthoi);

d2 = new JTextField();

d2.setHorizontalAlignment(SwingConstants.CENTER);

d2.setFont(new Font("Times New Roman", Font.PLAIN, 14));

d2.setColumns(10);

d2.setBounds(79, 241, 86, 35);

hthoi.add(d2);

JButton calhthoi = new JButton("Tính");

calhthoi.setFont(new Font("Times New Roman", Font.BOLD, 14));

calhthoi.setBounds(204, 180, 89, 35);

hthoi.add(calhthoi);

calhthoi.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//checkInput();

double d1HT = Double.valueOf(d1.getText());

double d2HT = Double.valueOf(d2.getText());

double SHT = 0.0;

SHT = (d1HT \* d2HT) / 2;

Shthoi.setText(Double.toString(SHT));

}

});

JLabel lblKtQu4 = new JLabel("Kết quả:");

lblKtQu4.setHorizontalAlignment(SwingConstants.CENTER);

lblKtQu4.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblKtQu4.setBounds(318, 125, 145, 48);

hthoi.add(lblKtQu4);

JButton delhthoi = new JButton("Xóa");

delhthoi.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

d1.setText("");

d2.setText("");

Shthoi.setText("");

}

});

delhthoi.setFont(new Font("Times New Roman", Font.BOLD, 14));

delhthoi.setBounds(204, 229, 89, 35);

hthoi.add(delhthoi);

//Hinh tron

JPanel htron = new JPanel();

tabbedPane.addTab("Hình tròn", null, htron, null);

htron.setLayout(null);

JLabel lblNewLabel5 = new JLabel("S = pi x r^2 hoặc S = (pi x d^2) / 4");

lblNewLabel5.setBounds(49, 23, 414, 48);

lblNewLabel5.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel5.setFont(new Font("Times New Roman", Font.BOLD, 20));

htron.add(lblNewLabel5);

JLabel lblACnhHnh5a = new JLabel("r: bán kính");

lblACnhHnh5a.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh5a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh5a.setBounds(154, 52, 193, 48);

htron.add(lblACnhHnh5a);

JLabel lblACnhHnh5b = new JLabel("d: đường kính");

lblACnhHnh5b.setHorizontalAlignment(SwingConstants.CENTER);

lblACnhHnh5b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblACnhHnh5b.setBounds(154, 80, 183, 48);

htron.add(lblACnhHnh5b);

JLabel lblNhpCnh5a = new JLabel("Nhập bán kính:");

lblNhpCnh5a.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh5a.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh5a.setBounds(49, 125, 145, 48);

htron.add(lblNhpCnh5a);

JLabel lblNhpCnh5b = new JLabel("Nhập đường kính:");

lblNhpCnh5b.setHorizontalAlignment(SwingConstants.CENTER);

lblNhpCnh5b.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNhpCnh5b.setBounds(49, 216, 145, 48);

htron.add(lblNhpCnh5b);

r = new JTextField();

r.setFont(new Font("Times New Roman", Font.PLAIN, 14));

r.setHorizontalAlignment(SwingConstants.CENTER);

r.setBounds(79, 165, 86, 35);

htron.add(r);

r.setColumns(10);

JTextArea Shtron = new JTextArea();

Shtron.setBounds(328, 165, 135, 128);

htron.add(Shtron);

d = new JTextField();

d.setHorizontalAlignment(SwingConstants.CENTER);

d.setFont(new Font("Times New Roman", Font.PLAIN, 14));

d.setColumns(10);

d.setBounds(79, 258, 86, 35);

htron.add(d);

JButton calhtron = new JButton("Tính");

calhtron.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//checkInput();

double SHTR = 0.0;

//Neu nhap d

if(r.getText().isEmpty() && !d.getText().isEmpty()) {

double dHT = Double.valueOf(d.getText());

SHTR = (pi \* Math.pow(dHT, 2)) / 4;

}

//Neu nhap r

if(d.getText().isEmpty() && !r.getText().isEmpty()) {

double rHT = Double.valueOf(r.getText());

SHTR = pi \* Math.pow(rHT, 2);

}

Shtron.setText(Double.toString(SHTR));

}

});

calhtron.setFont(new Font("Times New Roman", Font.BOLD, 14));

calhtron.setBounds(204, 180, 89, 35);

htron.add(calhtron);

JLabel lblKtQu5 = new JLabel("Kết quả:");

lblKtQu5.setHorizontalAlignment(SwingConstants.CENTER);

lblKtQu5.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblKtQu5.setBounds(318, 125, 145, 48);

htron.add(lblKtQu5);

JButton delhtron = new JButton("Xóa");

delhtron.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

r.setText("");

d.setText("");

Shtron.setText("");

}

});

delhtron.setFont(new Font("Times New Roman", Font.BOLD, 14));

delhtron.setBounds(204, 241, 89, 35);

htron.add(delhtron);

JLabel lblHoc = new JLabel("hoặc");

lblHoc.setHorizontalAlignment(SwingConstants.CENTER);

lblHoc.setFont(new Font("Times New Roman", Font.BOLD, 14));

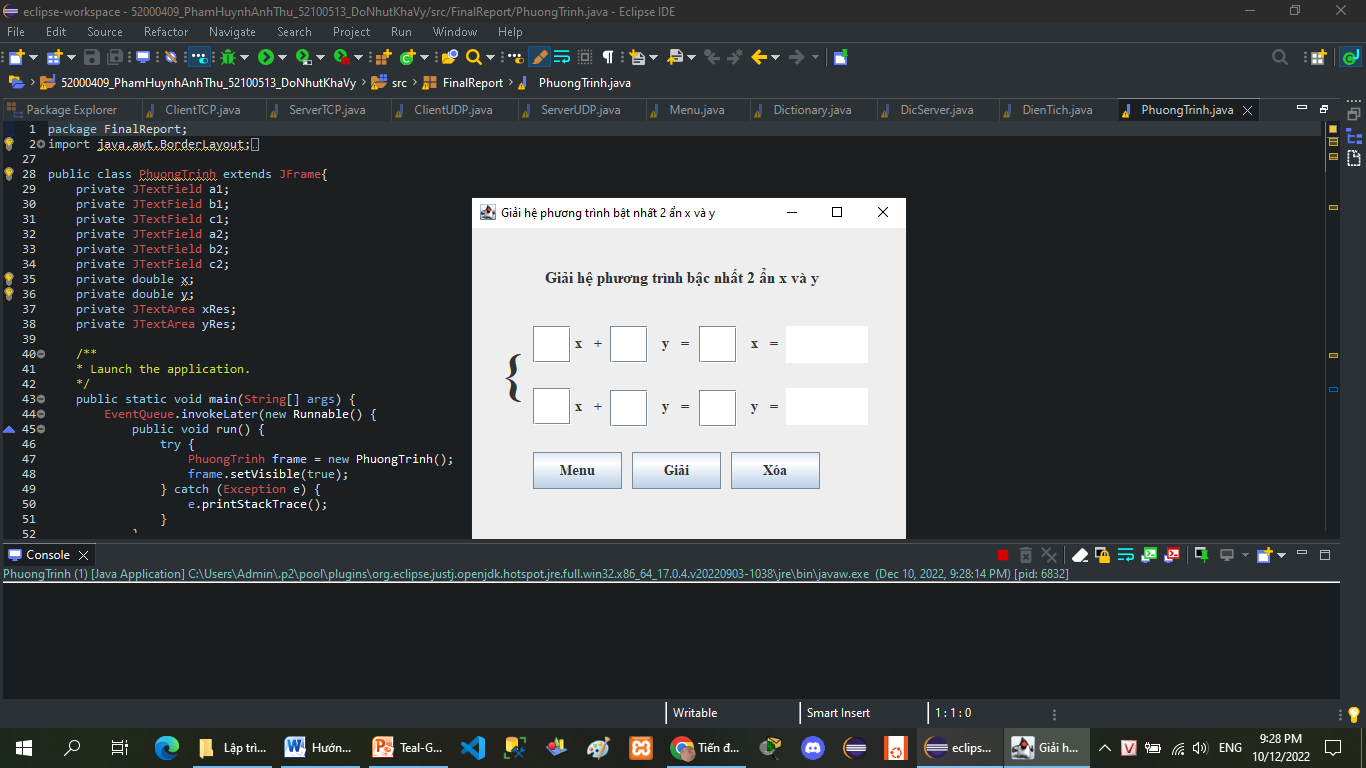
lblHoc.setBounds(49, 191, 145, 48);

htron.add(lblHoc);

}

}

3.3.3 CHƯƠNG TRÌNH GIẢI HỆ PHƯƠNG TRÌNH



**Code chương trình:**

***Giao diện client:***

package FinalReport;

import java.awt.BorderLayout;

import java.net.\*;

import java.util.Scanner;

import java.io.\*;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JTextArea;

import javax.swing.SwingConstants;

import javax.swing.JTabbedPane;

import javax.swing.event.AncestorListener;

import javax.swing.event.AncestorEvent;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.LayoutStyle.ComponentPlacement;

public class PhuongTrinh extends JFrame{

private JTextField a1;

private JTextField b1;

private JTextField c1;

private JTextField a2;

private JTextField b2;

private JTextField c2;

private double x;

private double y;

private JTextArea xRes;

private JTextArea yRes;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

PhuongTrinh frame = new PhuongTrinh();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\*

\*/

public void checkInput() {

if(a1.getText().isEmpty() || b1.getText().isEmpty() || c1.getText().isEmpty() || a2.getText().isEmpty() || b2.getText().isEmpty() || c2.getText().isEmpty()) {

Component fame = null;

JOptionPane.showMessageDialog(fame, "Vui lòng không nhập trống.", null, JOptionPane.WARNING\_MESSAGE);

}

else {

double n1, n2, n3, n4, n5, n6 = 0.0;

try {

n1 = Double.valueOf(a1.getText());

n2 = Double.valueOf(b1.getText());

n3 = Double.valueOf(c1.getText());

n4 = Double.valueOf(a2.getText());

n5 = Double.valueOf(b2.getText());

n6 = Double.valueOf(c2.getText());

} catch (NumberFormatException e) {

e.printStackTrace();

Component fame = null;

JOptionPane.showMessageDialog(fame, "Vui lòng nhập số.", null, JOptionPane.WARNING\_MESSAGE);

}

}

}

public PhuongTrinh() {

setTitle("Giải hệ phương trình bật nhất 2 ẩn x và y");

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 450, 350);

getContentPane().setLayout(null);

a1 = new JTextField();

a1.setBounds(61, 98, 38, 37);

a1.setColumns(10);

b1 = new JTextField();

b1.setBounds(138, 98, 38, 37);

b1.setColumns(10);

c1 = new JTextField();

c1.setBounds(227, 98, 38, 37);

c1.setColumns(10);

JLabel lblNewLabel = new JLabel(" x + ");

lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNewLabel.setBounds(99, 97, 49, 37);

getContentPane().setLayout(null);

getContentPane().add(a1);

getContentPane().add(b1);

getContentPane().add(c1);

getContentPane().add(lblNewLabel);

JLabel lblY = new JLabel(" y = ");

lblY.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblY.setBounds(186, 97, 49, 37);

getContentPane().add(lblY);

xRes = new JTextArea();

xRes.setBounds(314, 98, 82, 37);

getContentPane().add(xRes);

yRes = new JTextArea();

yRes.setBounds(314, 160, 82, 37);

getContentPane().add(yRes);

JLabel lblX = new JLabel(" x =");

lblX.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblX.setBounds(275, 97, 49, 37);

getContentPane().add(lblX);

JLabel lblY\_1 = new JLabel(" y =");

lblY\_1.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblY\_1.setBounds(275, 160, 49, 37);

getContentPane().add(lblY\_1);

JLabel lblNewLabel\_1 = new JLabel("Giải hệ phương trình bậc nhất 2 ẩn x và y");

lblNewLabel\_1.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblNewLabel\_1.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel\_1.setBounds(37, 28, 346, 45);

getContentPane().add(lblNewLabel\_1);

JButton solve = new JButton("Giải");

solve.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//Check input

checkInput();

//PT 1

double numA1 = Double.valueOf(a1.getText());

double numB1 = Double.valueOf(b1.getText());

double numC1 = Double.valueOf(c1.getText());

//PT 2

double numA2 = Double.valueOf(a2.getText());

double numB2 = Double.valueOf(b2.getText());

double numC2 = Double.valueOf(c2.getText());

try {

//result

//connect

Socket pt = new Socket("localhost", 2025);

//tao inputstream chua ket qua tu server

Scanner in = new Scanner(pt.getInputStream());

//tao outputstream chua word gui den server

PrintWriter out;

out = new PrintWriter(pt.getOutputStream(),true);

out.println(numA1);

out.println(numB1);

out.println(numC1);

out.println(numA2);

out.println(numB2);

out.println(numC2);

//lay result tu in

String x = in.nextLine();

String y = in.nextLine();

xRes.setText(x);

yRes.setText(y);

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

solve.setFont(new Font("Times New Roman", Font.BOLD, 14));

solve.setBounds(160, 224, 89, 37);

getContentPane().add(solve);

a2 = new JTextField();

a2.setColumns(10);

a2.setBounds(61, 160, 38, 37);

getContentPane().add(a2);

b2 = new JTextField();

b2.setColumns(10);

b2.setBounds(138, 162, 38, 37);

getContentPane().add(b2);

c2 = new JTextField();

c2.setColumns(10);

c2.setBounds(227, 162, 38, 37);

getContentPane().add(c2);

JLabel lblNewLabel\_2 = new JLabel(" x + ");

lblNewLabel\_2.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblNewLabel\_2.setBounds(99, 160, 49, 37);

getContentPane().add(lblNewLabel\_2);

JLabel lblY\_2 = new JLabel(" y = ");

lblY\_2.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblY\_2.setBounds(186, 160, 49, 37);

getContentPane().add(lblY\_2);

JLabel lblNewLabel\_3 = new JLabel("{");

lblNewLabel\_3.setFont(new Font("Times New Roman", Font.BOLD, 55));

lblNewLabel\_3.setBounds(29, 73, 22, 146);

getContentPane().add(lblNewLabel\_3);

JButton del = new JButton("Xóa");

del.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

a1.setText("");

b1.setText("");

c1.setText("");

a2.setText("");

b2.setText("");

c2.setText("");

xRes.setText("");

yRes.setText("");

}

});

del.setFont(new Font("Times New Roman", Font.BOLD, 14));

del.setBounds(259, 224, 89, 37);

getContentPane().add(del);

JButton menu = new JButton("Menu");

menu.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Menu frame = new Menu();

frame.setVisible(true);

}

});

menu.setFont(new Font("Times New Roman", Font.BOLD, 14));

menu.setBounds(61, 224, 89, 37);

getContentPane().add(menu);

}

}

***Server:***

package FinalReport;

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class PTServer {

public static void main(String[] args) {

try{

final ServerSocket svr = new ServerSocket(2025);

System.out.println("Server listening on port 2025...");

while(true){

Socket d = svr.accept();

new Thread(new PTRunnable(d)).start();

}

}catch(Exception ex){

ex.printStackTrace();

}

}

}

class PTRunnable implements Runnable{

Socket client = null;

PTRunnable(Socket client){

this.client = client;

}

@Override

public void run() {

try {

//lay input tu client

InputStream is = client.getInputStream();

Scanner in = new Scanner(is);

//tra ket qua

OutputStream os = client.getOutputStream();

PrintWriter out = new PrintWriter(os,true/\*auto-flush\*/);

while(in.hasNextLine()){

//1st

double a1 = Double.parseDouble(in.nextLine());

double b1 = Double.parseDouble(in.nextLine());

double c1 = Double.parseDouble(in.nextLine());

//2nd

double a2 = Double.parseDouble(in.nextLine());

double b2 = Double.parseDouble(in.nextLine());

double c2 = Double.parseDouble(in.nextLine());

//output

String result = "";

result = returnValue(a1, b1, c1, a2, b2, c2);

out.println(result);

}

} catch (Exception e) {

e.printStackTrace();

}

}

private String returnValue(double a1, double b1, double c1, double a2, double b2, double c2) {

String result = "";

double x = 0;

double y = 0;

String s1, s2, s3, s4 = "";

//Processing

double a = 0;

double b = 0;

double c = 0;

double ax = 0;

double by = 0;

//check case

double checkA = (double) a1 / a2;

double checkB = (double) b1 / b2;

double checkC = (double) c1 / c2;

//Case 1: He pt co 1 nghiem duy nhat: a1/a2 # b1/b2

if(checkA != checkB) {

//Cach 1: Cong dai so khi a1 + a2 = 0 || a1 - a2 = 0 || b1 + b2 = 0 || b1 - b2 = 0

if(a1 + a2 == 0 || a1 - a2 == 0 || b1 + b2 == 0 || b1 - b2 == 0) {

if(a1 + a2 == 0) {

a = 0;

b = b1 + b2;

c = c1 + c2;

y = (double) c / b;

x = (double) (c2 - b2\*y) / (a2);

}

if(a1 - a2 == 0) {

a = 0;

b = b1 - b2;

c = c1 - c2;

y = (double) c / b;

x = (double) (c2 - b2\*y) / (a2);

}

if(b1 + b2 == 0) {

a = a1 + a2;

b = 0;

c = c1 + c2;

x = (double) c / a;

y = (double) (c2 - a2\*x) / (b2);

}

if(b1 - b2 == 0) {

a = a1 - a2;

b = 0;

c = c1 - c2;

x = (double) c / a;

y = (double) (c2 - a2\*x) / (b2);

}

result += Double.toString(x) + "\n" + Double.toString(y);

}

//Cach 2: Dung phuong phap the, dat x hoac y.

//Neu dat x: x = (c - by) / a

//Neu dat y: y = (c - ax) / b

else if(a1 + a2 != 0 && a1 - a2 != 0 && b1 + b2 != 0 && b1 - b2 != 0) {

if(a1 + a2 != 0 && a1 - a2 != 0) {

if(a1 < a2) {

y = (double) (c2 \* a1 - a2 \* c1) / (-a2 \* b1 + a1 \* b2);

x = (double) (c1 - b1 \* y) / a1;

}

else {

y = (double) (c1 \* a2 - a1 \* c2) / (-a1 \* b2 + a2 \* b1);

x = (double) (c2 - b2 \* y) / a2;

}

}

result += Double.toString(x) + "\n" + Double.toString(y);

}

}

//Case 2: He pt vo nghiem: a1/a2 = b1/b2 # c1/c2

else if(checkA == checkB && checkB != checkC || checkA != checkC) {

s1 = "Không có nghiệm.";

s2 = "Không có nghiệm.";

result += s1 + "\n" + s2;

}

//Case 3: He pt co vo so nghiem: a1/a2 = b1/b2 = c1/c2

else if(checkA == checkB && checkB == checkC && checkA == checkC) {

s3 = "Vô số nghiệm.";

s4 = "Vô số nghiệm.";

result += s3 + "\n" + s4;

}

//

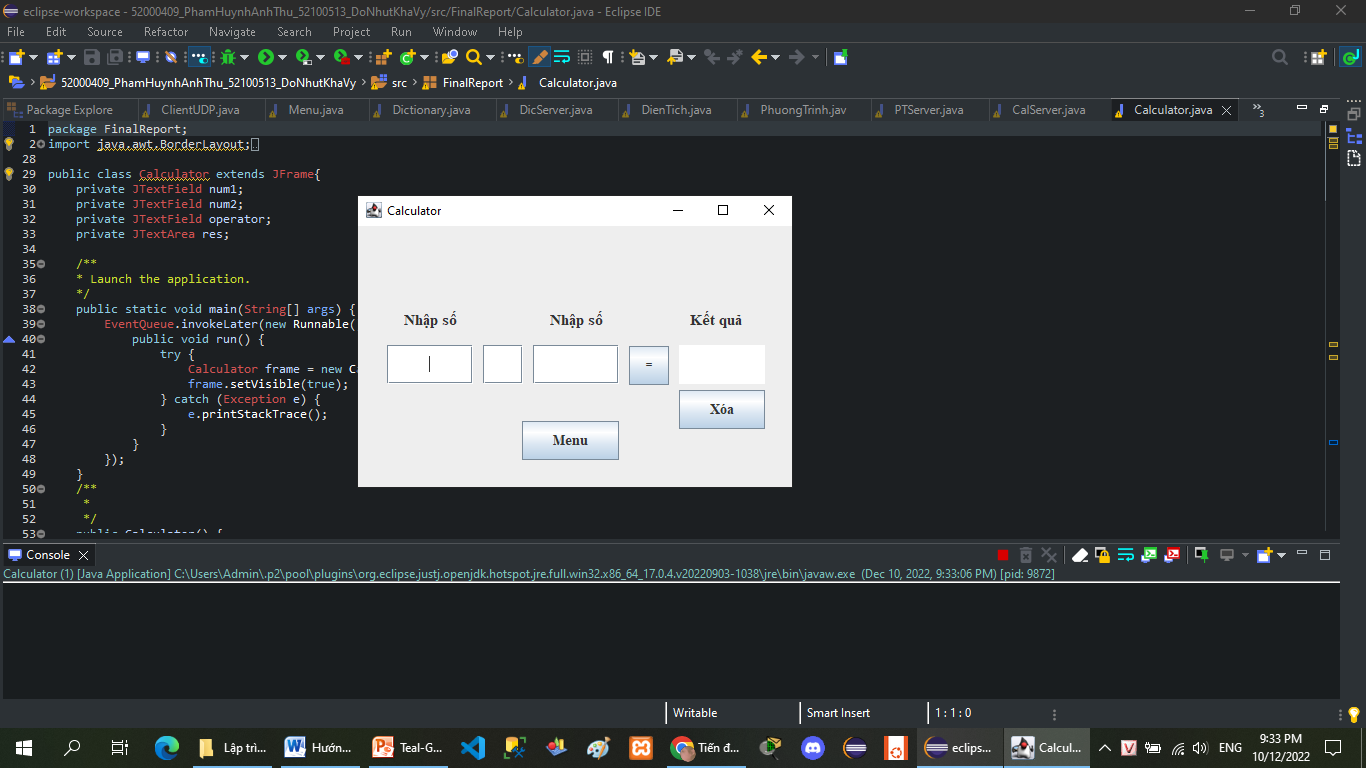
System.out.println(result);

return result;

}

}

3.3.4 CHƯƠNG TRÌNH TÍNH TOÁN SỐ HỌC



**Code chương trình:**

***Giao diện client:***

package FinalReport;

import java.awt.BorderLayout;

import java.awt.Component;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.PrintWriter;

import java.io.UnsupportedEncodingException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.Socket;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.util.Scanner;

import java.awt.event.ActionEvent;

import javax.swing.JTextArea;

import javax.swing.SwingConstants;

public class Calculator extends JFrame{

private JTextField num1;

private JTextField num2;

private JTextField operator;

private JTextArea res;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Calculator frame = new Calculator();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\*

\*/

public Calculator() {

setTitle("Calculator");

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("Nhập số ");

lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 15));

lblNewLabel.setBounds(46, 80, 69, 28);

getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("Nhập số ");

lblNewLabel\_1.setFont(new Font("Times New Roman", Font.BOLD, 15));

lblNewLabel\_1.setBounds(192, 80, 69, 28);

getContentPane().add(lblNewLabel\_1);

num1 = new JTextField();

num1.setHorizontalAlignment(SwingConstants.CENTER);

num1.setBounds(29, 119, 86, 39);

getContentPane().add(num1);

num1.setColumns(10);

num2 = new JTextField();

num2.setHorizontalAlignment(SwingConstants.CENTER);

num2.setColumns(10);

num2.setBounds(175, 119, 86, 39);

getContentPane().add(num2);

operator = new JTextField();

operator.setHorizontalAlignment(SwingConstants.CENTER);

operator.setBounds(125, 119, 40, 39);

getContentPane().add(operator);

operator.setColumns(10);

JLabel lblKtQu = new JLabel("Kết quả");

lblKtQu.setFont(new Font("Times New Roman", Font.BOLD, 15));

lblKtQu.setBounds(332, 80, 60, 28);

getContentPane().add(lblKtQu);

JButton cal = new JButton("=");

cal.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

//check rong

/\*if(num1.getText() == null || num2.getText() == null || operator.getText() == null) {

JOptionPane.showMessageDialog(null, "Vui lòng nhập phép tính '+', '-', 'x', '\*', ':' hoặc '/'");

res.setText("");

}\*/

double n1 = Double.valueOf(num1.getText());

double n2 = Double.valueOf(num2.getText());

double n = 0.0;

String op = operator.getText();

String result = "";

//connect

Socket cal = new Socket("localhost", 2023);

//tao inputstream chua ket qua tu server

Scanner in = new Scanner(cal.getInputStream());

//tao outputstream chua word gui den server

PrintWriter out = new PrintWriter(cal.getOutputStream(),true);

out.println(n1);

out.println(n2);

out.println(op);

//lay result tu in

result = in.nextLine();

res.setText(result);

} catch (Exception e1) {

e1.printStackTrace();

}

}

});

cal.setFont(new Font("Times New Roman", Font.BOLD, 10));

cal.setBounds(271, 120, 40, 39);

getContentPane().add(cal);

res = new JTextArea();

res.setColumns(2);

res.setFont(new Font("Times New Roman", Font.PLAIN, 14));

res.setBounds(321, 119, 86, 39);

getContentPane().add(res);

JButton back = new JButton("Menu");

back.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

Menu frame = new Menu();

frame.setVisible(true);

}

});

back.setFont(new Font("Times New Roman", Font.BOLD, 14));

back.setBounds(164, 195, 97, 39);

getContentPane().add(back);

JButton del = new JButton("Xóa");

del.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

num1.setText("");

num2.setText("");

operator.setText("");

res.setText("");

}

});

del.setFont(new Font("Times New Roman", Font.BOLD, 14));

del.setBounds(321, 164, 86, 39);

getContentPane().add(del);

}

}

***Server:***

package FinalReport;

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class CalServer {

public static void main(String[] args) {

try{

final ServerSocket svr = new ServerSocket(2023);

System.out.println("Server listening on port 2023...");

while(true){

Socket d = svr.accept();

new Thread(new CalRunnable(d)).start();

}

}catch(Exception ex){

ex.printStackTrace();

}

}

}

class CalRunnable implements Runnable{

Socket client = null;

CalRunnable(Socket client){

this.client = client;

}

@Override

public void run() {

try {

//lay input tu client

InputStream is = client.getInputStream();

Scanner in = new Scanner(is);

//tra ket qua

OutputStream os = client.getOutputStream();

PrintWriter out = new PrintWriter(os,true/\*auto-flush\*/);

while(in.hasNextLine()){

double a = Double.parseDouble(in.nextLine());

double b = Double.parseDouble(in.nextLine());

String op = in.nextLine();

double result = returnValue(a, b, op);

String res = Double.toString(result);

out.println(res);

}

} catch (Exception e) {

e.printStackTrace();

}

}

private double returnValue(double a, double b, String op) {

double result = 0.0;

if(op.equals("+")) {

result = a + b;

}

else if(op.equals("-")) {

result = a - b;

}

else if(op.equals("x") || op.equals("\*")) {

result = a \* b;

}

else if(op.equals(":") || op.equals("/")) {

result = (double) a / b;

}

return result;

}

}

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. <https://vovanhai.wordpress.com/java/l%E1%BA%ADp-trinh-java-nang-cao/networking/l%E1%BA%ADp-trinh-socket-t%E1%BA%A1o-%E1%BB%A9ng-d%E1%BB%A5ng-calculator/>

**Tiếng Anh**

1. <https://www.javatpoint.com/socket-programming>
2. <https://github.com/KvnGriffin/Chat-Client-Server/blob/master/UDPServer.java>
3. <https://stackoverflow.com/questions/68325685/udp-client-server-calulator-with-gui>
4. <https://www.codejava.net/java-se/networking/java-udp-client-server-program-example>
5. <https://www.youtube.com/watch?v=5f6xyUDNeYc&t=2192s>
6. <https://www.youtube.com/watch?v=-xKgxqG411c>