

# JAVA SWING CHAT GUI

นาย บุญญฤทธิ์ เจิดรุจิกุล รหัสนิสิต 6530300317

วิชา Programming Fundamentals II รหัสวิชา 03603112-65

อาจารย์ผู้สอน กุลวดี สมบูรณ์วิวัฒน์ ปีการศึกษา 2565

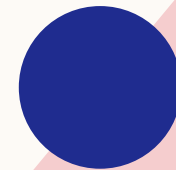
# SECTION

Introduction & Reason

UI

Algorithm

Installation

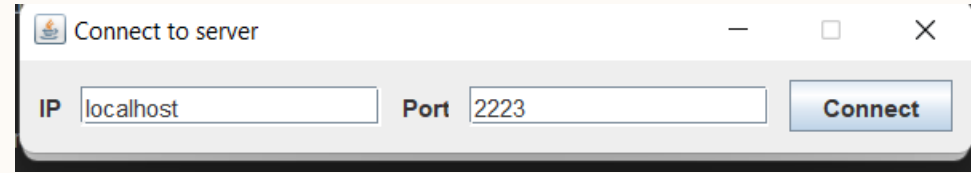


# WHY THIS PROJECT

เหตุผลที่เลือก project นี้เพราะเป็น project ที่สามารถเอาไปพัฒนาต่อเพิ่มเติมได้หลายๆ ส่วน เช่น การ Transfer file, Chatbot ที่ใช้ตอบกลับ, ใช้โปรแกรม ngrok

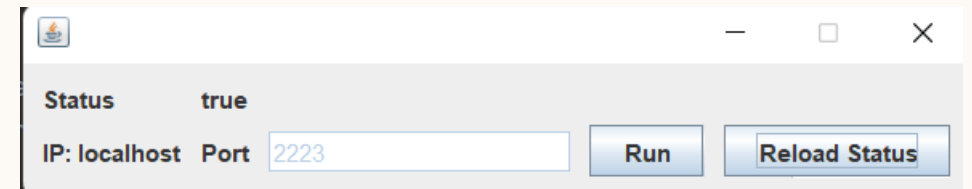
# CHAT UI

Component ของ Chat UI ก็จะมี textField, button , textArea, และ label ในส่วนของ textField จะเป็นตัวรับ input ของ user และนำไปแสดงใน textArea ก่อนที่เข้าทำงานในตัวส่วน ChatClient จะมีคลาส Connect ที่เป็น gui แสดงขึ้น ในส่วนของคลาสนี้ component textField จะทำการรับ port และ ip มาไปเชื่อมกับ socket



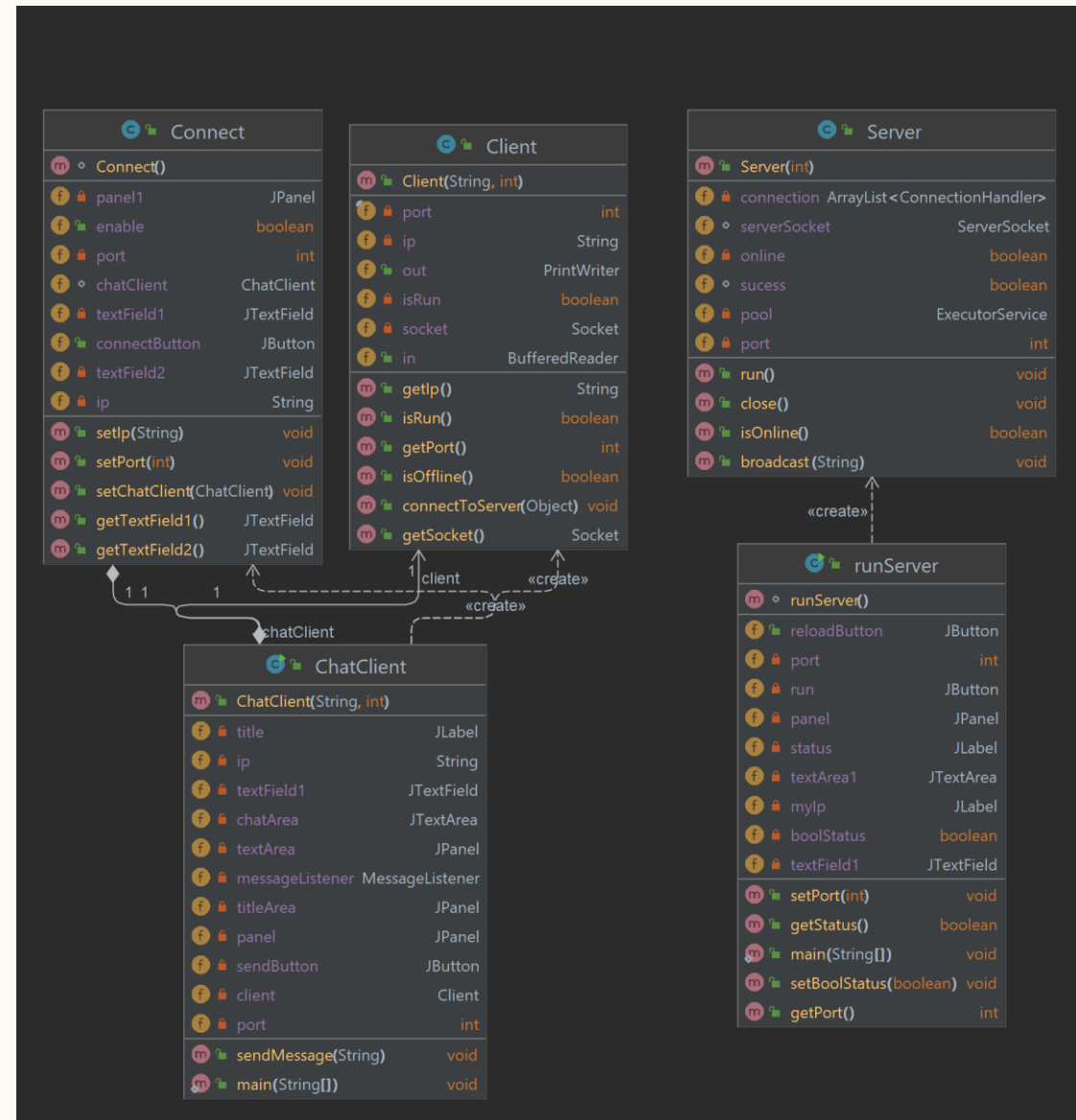
# SERVER UI

Component ของ UI server จะมี label, textfield และ button ในส่วนขอตัว button จะมี 2 ปุ่ม button run จะทำการสร้างเซิร์ฟเวอร์ และ button reload จะทำการเช็คการเชื่อมต่อของเซิร์ฟเวอร์ว่ายังทำงานอยู่หรือไม่ textfield ก็จะทำหน้าที่ใส่ port ของเซิร์ฟเวอร์



# UML DIAGRAM

- Class Connect - เป็นคลาส gui รับ input ของ textfield และ ส่ง ip กับ port ที่ user กรอกมาไปให้คลาส Chalclient ให้เชื่อมต่อ socket
- Class Client - เป็น class ที่ใช้เชื่อมต่อ socket ของ server และรับ input ของ user แล้วเอาไปแสดงผลในคลาส Chatclient
- Class Server - เป็นคลาสที่ทำหน้าที่เป็น Host ให้ตัว socket client เชื่อมต่อ
- Class Chatclient - เป็นคลาส gui ที่ใช้ chat และกับ socket ของคลาส Client
- Class runServer - เป็นคลาส gui รับ input ของ textfield และ ส่ง ip กับ port ที่ user กรอกมาไปเชื่อมต่อกับ socket ของคลาส Server



# ALGORITHM

## Class Client

ในคลาสนี้จะมี constructure จะรับค่า parameters มา set port และ ip ลงใน ตัวแปรทั้งสอง เพื่อนำ ip และ port ไปใช้ใน method connectToServer เพื่อเชื่อมต่อ socket และรันใน thread

```
package com.components.system;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.lang.reflect.Constructor;
import java.net.ConnectException;
import java.net.Socket;
import java.net.UnknownHostException;

public class Client {
    private String ip;
    private final int port;
    private Socket socket;
    public BufferedReader in;
    public PrintWriter out;

    public boolean isRun = true;

    public Client(String ip, int port){
        this.ip = ip;
        this.port = port;
    }

    public void connectToServer(Object obj){
        try {

            socket = new Socket(ip, port);
            in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            out = new PrintWriter(socket.getOutputStream(), true);

            Thread thread = new Thread((Runnable) obj);
            thread.start();

        } catch (IOException e) {
            this.isRun = false;
        }
    }

    public int getPort() {
        return port;
    }

    public String getIp() {
        return ip;
    }
}
```

# ALGORITHM

## Class Sever [1]

ในคลาสนี้จะมี constructure จะรับค่า parameters มา set port เพื่อเชื่อมกับ socket และ เก็บ list ของการเชื่อมต่อ ในส่วนของ method run จะรับ port ที่ set ไว้ เชื่อมกับ socket สร้าง thread และคอยเก็บ connection ของ client method close จะทำหน้าที่คอยปิด connection ทั้งหมดเมื่อเซิร์ฟเวอร์ปิดลง

และ method broadcast คือ method ที่ทำการส่งข้อความหรือคำสั่งจาก sever ไปยัง client

```
package com.components.system;

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.ArrayList;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class Server implements Runnable {

    private int port;
    private ArrayList<ConnectionHandler> connection;

    ServerSocket serverSocket;
    boolean success;
    private ExecutorService pool;

    private boolean online;

    public boolean isOnline() {
        return online;
    }

    public Server(int port){
        this.port = port;
        connection = new ArrayList<>();
        success = false;
    }

    @Override
    public void run() {

        try {
            serverSocket = new ServerSocket(2223);
            pool = Executors.newCachedThreadPool();
            this.online = !serverSocket.isClosed();
            while (!success) {
                Socket client = serverSocket.accept();

                ConnectionHandler handler = new ConnectionHandler(client);
                connection.add(handler);
                pool.execute(handler);
            }
        } catch (IOException e) {
            close();
        }
    }

    public void close() {
        success = true;
        try{
            if(!serverSocket.isClosed()){
                serverSocket.close();
                for(ConnectionHandler channel:connection){
                    channel.close();
                }
            }
        } catch (IOException e){

        }
    }

    public void broadcast(String message){
        for(ConnectionHandler channel: connection){
            if(channel != null){
                channel.sendMessage(message);
            }
        }
    }
}
```



# ALGORITHM

## Class Sever [2]

ClassNest ConnectionHandler จะมี method sendMessage และรับ parameters เป็น string และส่งข้อความ ไปให้ client method run เก็บ connection ของเซิร์ฟเวอร์ ตลอดเวลาด้วยการลูป และคอยรับคำสั่งจาก client เช่น !exit เพื่อ disconnect

```
class ConnectionHandler implements Runnable{
    private Socket client;
    private BufferedReader in;
    private PrintWriter out;
    private String username;
    public ConnectionHandler(Socket client){
        this.client = client;
    }
    @Override
    public void run() {
        try{
            out = new PrintWriter(client.getOutputStream(), true );
            in = new BufferedReader(new InputStreamReader(client.getInputStream()));
            out.println("Please enter your username: ");
            username = in.readLine();
            System.out.println(username + " is connected!");
            broadcast(username + " join the chat!");
            String message;
            while ((message = in.readLine()) != null){

                if(message.equals("!exit")){
                    broadcast(username+" left the chat!");
                    close();

                }else{
                    broadcast(String.format("%s : %s", username, message));
                }
            }
        } catch (IOException e){
            broadcast(String.format("runServer was terminated"));
            close();
        }
    }

    public void sendMessage(String message){
        out.println(message);
    }
    public void close(){
        try{
            in.close();
            out.close();
            if(!client.isClosed()){
                client.close();
            }
        } catch (IOException e){
            close();
        }
    }
}
```

# ALGORITHM

## Class ChatClient [1]

คลาส ChatClient เป็นคลาส gui ที่มี constructor ที่รับ parameters ip และ port และทำการ set แล้ว inner Class MessageListener จะมี method run ที่ implements Runnable จะทำการเอาความข้อความที่ user กรอกไปโชว์ใน textArea และมี method sendMessage ส่งข้อความ

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.io.*;

import com.components.system.Client;

public class ChatClient extends JFrame{
    private JTextField textField1;
    private JButton sendButton;
    private JPanel panel;
    private JPanel titleArea;
    private JPanel textArea;
    private JLabel title;
    private JTextArea chatArea;
    private Client client;

    private String ip;
    private int port;

    private MessageListener messageListener;

    private class MessageListener implements Runnable {
        @Override
        public void run() {
            try {
                String message;
                while ((message = client.in.readLine()) != null) {
                    chatArea.append(message + "\n");
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public void sendMessage(String message){
        if(!message.isEmpty()){
            if(message.equals("!exit")){
                setVisible(false);
            }
            client.out.println(message);
            textField1.setText("");
        }
    }
}
```

# ALGORITHM

## Class ChatClient [2]

และมี main method ที่ เรียกใช้คลาส gui connect เพื่อ get ip กับ port เพื่อที่จะ run ในตัว gui และเชื่อม socket ของ class chatClient

```
public ChatClient(String ip, int port) {

    this.ip = ip;
    this.port = port;

    client = new Client(ip, port);
    MessageListener messageListener = new MessageListener();
    client.connectToServer(messageListener);
    titleArea.setPreferredSize(new Dimension(95, 50));
    title.setText("Chat GUI");
    title.setForeground(new Color(255, 255, 255));
    title.setPreferredSize(new Dimension(75, 28));

    chatArea.setEditable(false);

    TextArea.setPreferredSize(new Dimension(95, 50));
    textField1.setPreferredSize(new Dimension(75, 35));
    sendButton.setPreferredSize(new Dimension(75, 35));

    setContentPane(panel);

    validate();
    repaint();
    setResizable(false);

    setSize(533, 571);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setVisible(true);

    sendButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            sendMessage(textField1.getText());
            System.out.println(textField1.getText());

        }
    });

}

public static void main(String[] args) {

    Connect connect = new Connect();
    connect.connectButton.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            if(!connect.getTextField1().getText().isBlank() &&
            !connect.getTextField2().getText().isBlank() && connect.getTextField2().getText().matches("[0-9]+")){
                connect.enable = true;
                connect.setIp(connect.getTextField1().getText());
                connect.setPort(Integer.parseInt(connect.getTextField2().getText()));
                ChatClient chatClient = new ChatClient(connect.getTextField1().getText(),
                Integer.parseInt(connect.getTextField2().getText()));

                connect.dispose();

            }else{
                JOptionPane.showMessageDialog(null, "Something went wrong", "Warning",
                JOptionPane.ERROR_MESSAGE);
            }
        }
    });

}

}
```

# ALGORITHM

## Class Connect

คลาส Connect จะสร้าง component และ layout ในโปรแกรมจะมี getter setter ที่คอย set field และ set layout

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Connect extends JFrame{
    private JTextField textField1;
    private JPanel panel1;
    public JButton connectButton;
    private JTextField textField2;
    private String ip;
    private int port;
    public boolean enable = false;
    ChatClient chatClient;

    public void setChatClient(ChatClient chatClient) {
        this.chatClient = chatClient;
    }

    public JTextField getTextField1() {
        return textField1;
    }

    public JTextField getTextField2() {
        return textField2;
    }

    public void setIp(String ip) {
        this.ip = ip;
    }

    public void setPort(int port) {
        this.port = port;
    }

    Connect(){
        setTitle("Connect to server");
        panel1.setBorder(BorderFactory.createCompoundBorder(
            BorderFactory.createEmptyBorder(5, 5, 5, 5),
            BorderFactory.createEmptyBorder(5, 5, 5, 5)));
        setContentPane(panel1);
        validate();
        repaint();
        setResizable(false);
        pack();
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}
```

# ALGORITHM

## Class RunSever

คลาส RunServer จะมี getter setter ที่คอย set field, set layout และเชื่อมกับ socket server ใน method main

```
import com.components.system.Server;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RunServer extends JFrame {
    private JButton run;
    private JTextField textField1;
    private JTextArea textArea1;
    private JLabel myIp;
    private JPanel panel;
    private JLabel status;
    private JButton reloadButton;
    private boolean boolStatus = false;
    private int port;

    public void setBoolStatus(boolean boolStatus) {
        this.boolStatus = boolStatus;
    }

    public int getPort() {
        return port;
    }

    public void setPort(int port) {
        this.port = port;
    }

    public boolean getStatus() {
        return boolStatus;
    }

    RunServer(){
        panel.setBorder(BorderFactory.createCompoundBorder(
            BorderFactory.createEmptyBorder(5, 5, 5, 5),
            BorderFactory.createEmptyBorder(5, 5, 5, 5)));
        myIp.setText("IP: localhost");
        setContentPane(panel);

        validate();
        repaint();
        setResizable(false);

        pack();
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setVisible(true);

        status.setText(String.valueOf(boolStatus));

        run.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                if(textField1.getText() != null){
                    setPort(Integer.parseInt(textField1.getText()));
                    textField1.setEnabled(false);
                }
            }
        });
    }

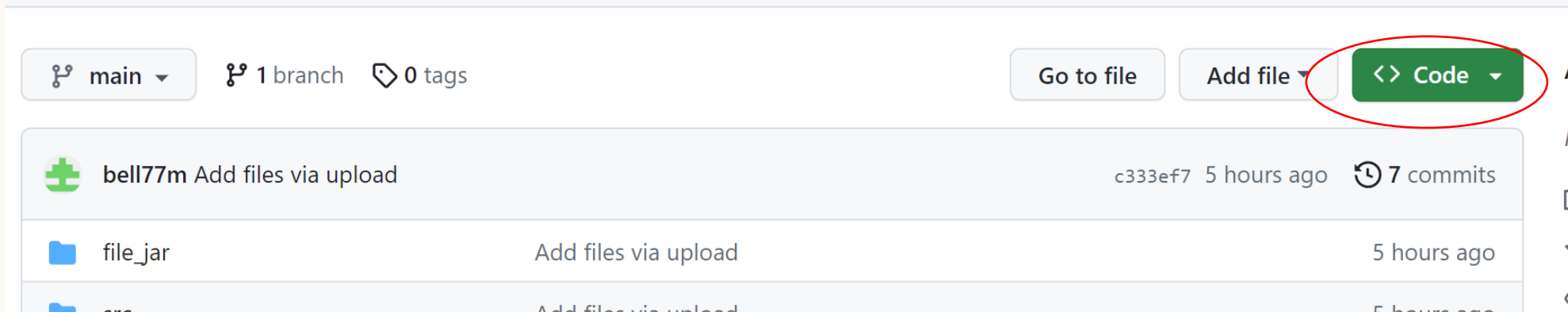
    public static void main(String[] args) {
        RunServer runServer = new RunServer();
        Server server = new Server(runServer.getPort());

        runServer.reloadButton.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                runServer.setBoolStatus(server.isOnline());
                runServer.status.setText(String.valueOf(runServer.getStatus()));
            }
        });
        server.run();
    }
}
```

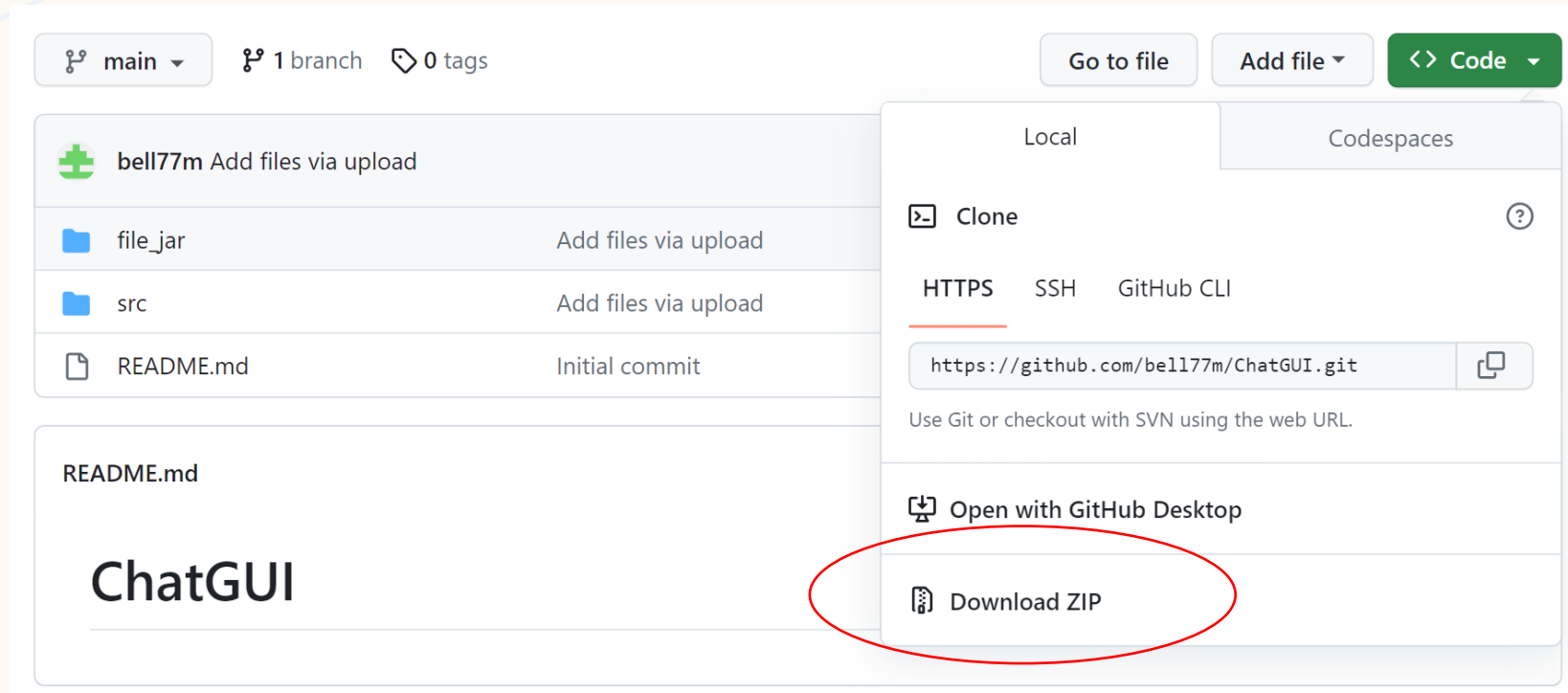
# INSTALLATION

1. เข้า website -> <https://github.com/bell77m/ChatGUI>



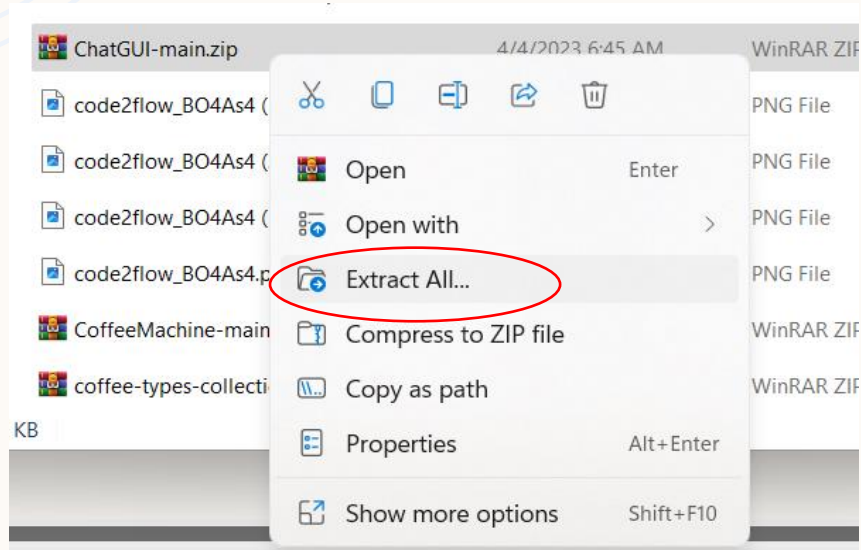
2. Click ที่ปุ่ม code

# INSTALLATION



3. Click ที่ปุ่ม Download ZIP เพื่อ download file (source code + jar) ของโปรแกรม

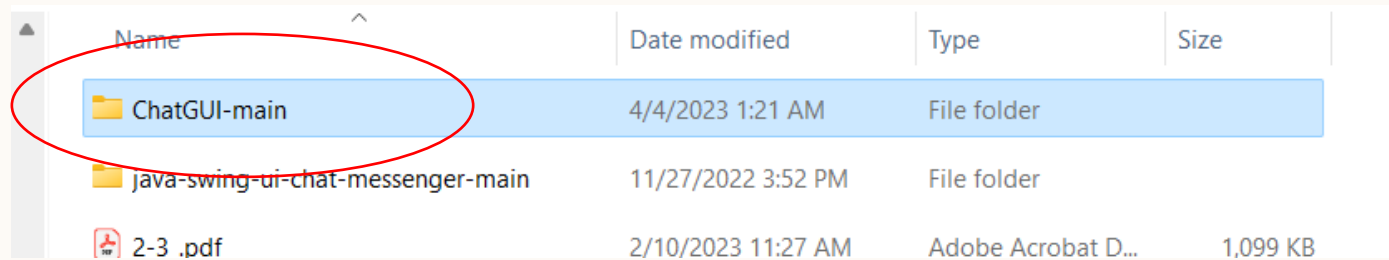
# INSTALLATION



4. เมื่อได้ไฟล์ zip มาแล้วก็ทำการ extract file ออกมา






# INSTALLATION



Name	Date modified	Type	Size
ChatGUI-main	4/4/2023 1:21 AM	File folder	
java-swing-ui-chat-messenger-main	11/27/2022 3:52 PM	File folder	
2-3 .pdf	2/10/2023 11:27 AM	Adobe Acrobat D...	1,099 KB



5. แลกด folder ที่ทำการ extract file มา

# INSTALLATION

 file_jar	4/4/2023 1:21 AM	File folder	
 src	4/4/2023 1:21 AM	File folder	
 README.md	4/4/2023 1:21 AM	Markdown Source ...	1 KB



6. กดไปที่ file\_jar

# INSTALLATION

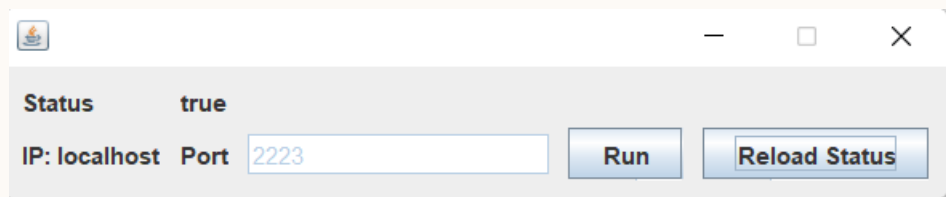
 Client.jar	4/4/2023 1:21 AM	Executable Jar File	168 KB
 Server.jar	4/4/2023 1:21 AM	Executable Jar File	168 KB

7. แล้วก็จะเจอกับโปรแกรม client.jar และ server.jar สามารถใช้ได้โดย  
(ก่อนที่จะใช้งานโปรแกรมต้องลง java ก่อน -> <https://www.java.com/en/> )

# HOW TO USE IT?



 Client.jar	4/4/2023 1:21 AM	Executable Jar File	168 KB
 Server.jar	4/4/2023 1:21 AM	Executable Jar File	168 KB

1. Double click ที่ Server.jar ไฟล์

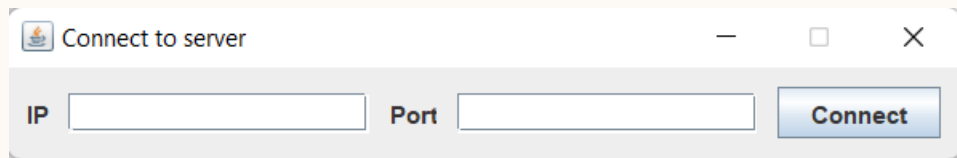


2. กรอก port ลงใน textField แล้วกดรัน

# HOW TO USE IT?

 Client.jar	4/4/2023 1:21 AM	Executable Jar File	168 KB
 Server.jar	4/4/2023 1:21 AM	Executable Jar File	168 KB

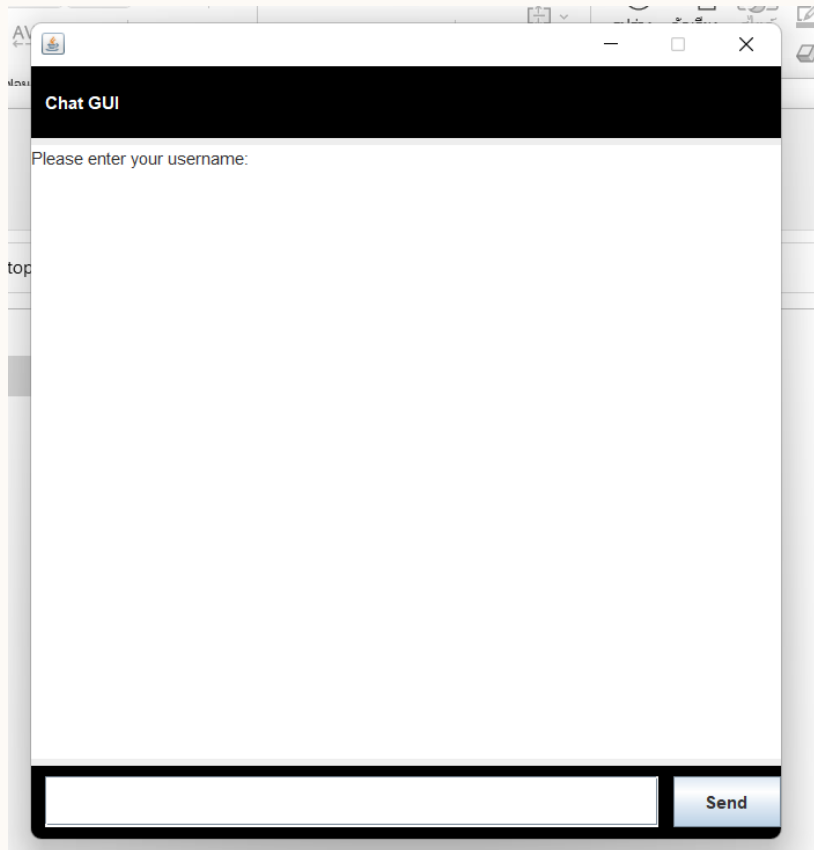
1. Double click ที่ Client.jar ไฟล์



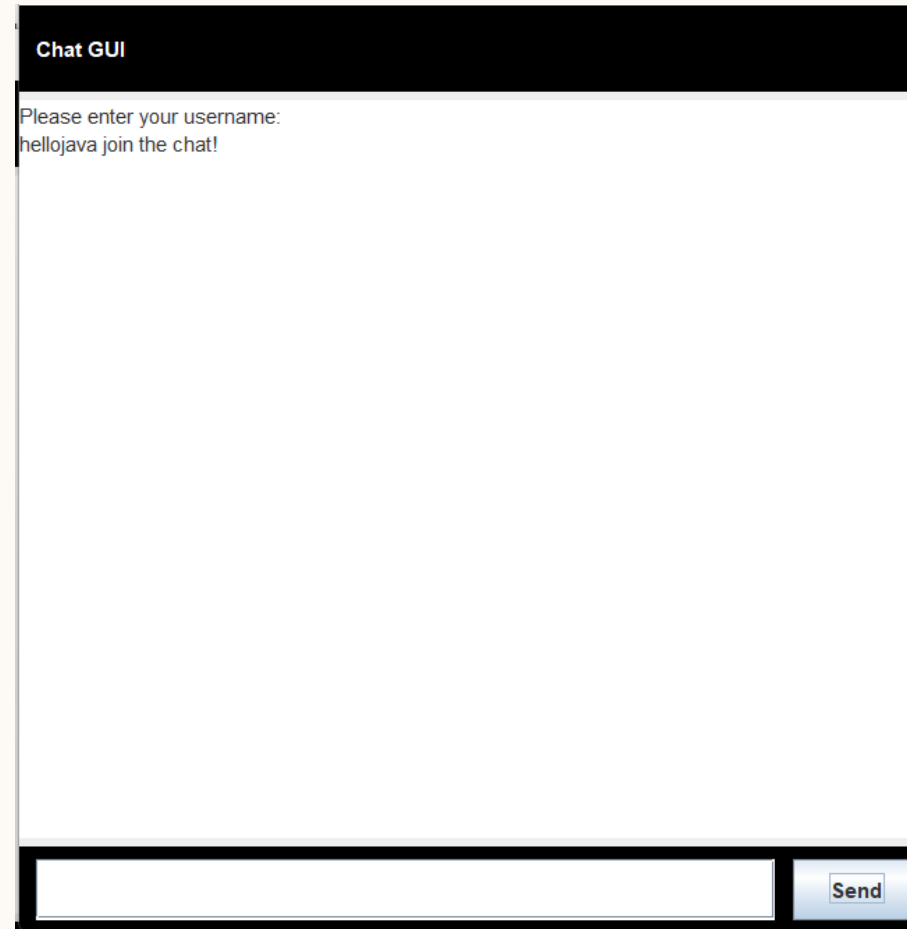
A dialog box titled "Connect to server" with a standard Windows window frame. It contains two text input fields: one labeled "IP" and another labeled "Port". To the right of these fields is a blue button labeled "Connect".

2.กรอก ip กับ port ให้ตรงกับเซิร์ฟเวอร์ที่เราจะเชื่อมต่อ ลงใน textfield แล้วกด connect

# HOW TO USE IT?

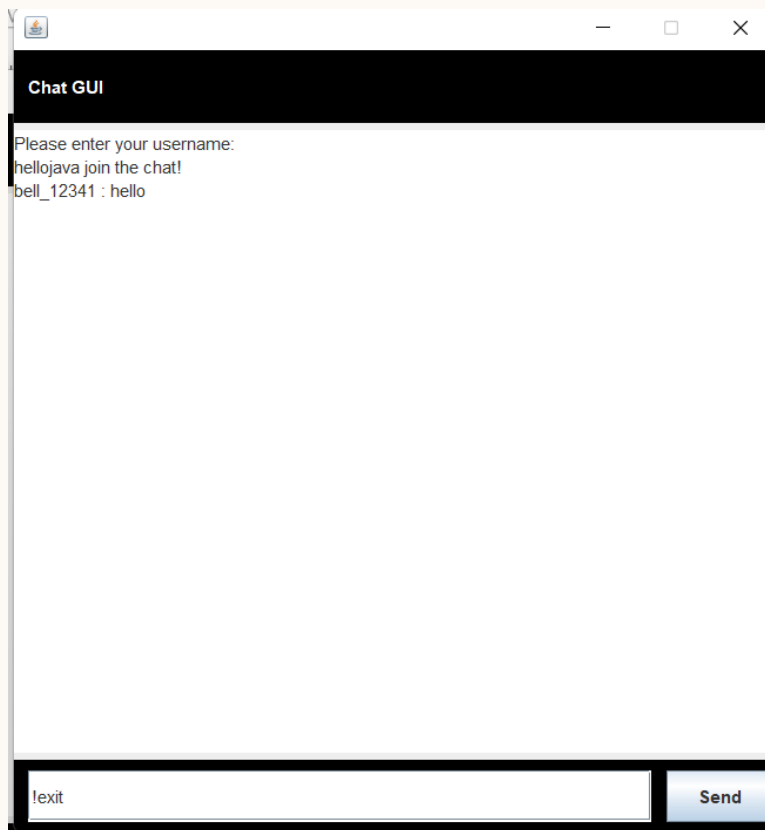


3.ก่อนที่จะแชทได้เราต้องกรอก username ที่จะโชว์ใน display แล้วกด send



4.แล้วพอขึ้น username ของเราแล้วก็ข้อความ join chat ก็ใช้ได้เลย

# HOW TO USE IT?



5. แล้วพอพิมพ์ !exit ก็จะได้ disconnect ออกจากแชท แล้วก็ขึ้นข้อความ left the chat!

# THANK YOU

Github: <https://github.com/bell77m/ChatGUI>

Video : <https://www.youtube.com/watch?v=mTzNyXlJDz8>