VD1

Server

import java.io.IOException;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

public class VD1Server {

    public static void main(String[] args)  {

        try{

            ServerSocket serverSocket = new ServerSocket(12345);

            System.out.println("Server started and listening on port 12345.....");

            while (true) {

                Socket clientSocket = serverSocket.accept();

                System.out.println("Client connected: " + clientSocket);

                Thread clientThread= new Thread( new ClientHandler(clientSocket));

                clientThread.start();

            }

        } catch (IOException e)

        {

            e.printStackTrace();

        }

    }

}

 class ClientHandler implements Runnable{

    private Socket clientSocket;

     public ClientHandler (Socket clientSocket){

        this.clientSocket=clientSocket;

     }

     @Override

     public void run(){

        try{

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

            out.println("Hello from server");

            clientSocket.close();

            System.out.println("Client disconnected: " + clientSocket);

        }catch (IOException e)

        {

            e.printStackTrace();

        }

     }

 }

Client

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class VD1Client {

    public static void main(String [] args) {

        try{

            Socket socket = new Socket("localhost",12345);

            System.out.println("Connected to server: " + socket);

            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

            String message = in.readLine();

            System.out.println("Received message from server: " + message);

            socket.close();

            System.out.println("Disconnected from server: " + socket);

        }catch (IOException e){

            e.printStackTrace();

        }

    }

}

A screenshot of a computer

Description automatically generated

VD2

Server

import java.io.IOException;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

public class VD2Server {

    public static void main (String [] args){

        try{

            ServerSocket serverSocket = new ServerSocket(12345);

            System.out.println("Server started and listening on port 12345.....");

            while (true) {

                Socket clientSocket = serverSocket.accept();

                System.out.println("Client connected: " + clientSocket);

                Thread clienThread = new ClientHandler(clientSocket);

                clienThread.start();

            }

        } catch (IOException e)

        {

            e.printStackTrace();

        }

    }

}

class ClientHandler extends Thread{

    private Socket clientSocket;

    public ClientHandler(Socket clientSocket){

        this.clientSocket=clientSocket;

    }

    @Override

    public void run(){

        try{

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

            out.println("Hello from server");

            clientSocket.close();

                System.out.println("Client disconnected: " + clientSocket);

        }catch(IOException e)

        {

            e.printStackTrace();

        }

    }

}

Client

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class VD2Client {

    public static void main(String [] args) {

        try{

            Socket socket = new Socket("localhost",12345);

            System.out.println("Connected to server: " + socket);

            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

            String message = in.readLine();

            System.out.println("Received message from server: " + message);

            socket.close();

            System.out.println("Disconnected from server: " + socket);

        }catch (IOException e){

            e.printStackTrace();

        }

    }

}

A screenshot of a computer screen

Description automatically generated

VD3

Server

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;;

public class VD3Server {

    public static void main (String [] args)

    {

        ExecutorService executorService = Executors.newFixedThreadPool(10);

        try{

            ServerSocket serverSocket = new ServerSocket(12345);

            System.out.println("Server started and listening on port 12345");

            while (true) {

                Socket clientSocket = serverSocket.accept();

                System.out.println("Client connected: " + clientSocket);

                executorService.execute(new ClientHandler(clientSocket));

            }

        }catch (IOException e)

        {

            e.printStackTrace();

        }finally{

            executorService.shutdown();

        }

    }

}

class ClientHandler implements Runnable{

    private Socket clientSocket;

    public ClientHandler (Socket clientSocket){

        this.clientSocket=clientSocket;

    }

    @Override

    public void run(){

        try{

            BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

            String request=in.readLine();

            System.out.println("Received request from client: " + request);

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

            out.println("Response from server: " + request);

            clientSocket.close();

            System.out.println("Client disconnected: " + clientSocket);

        } catch (IOException e)

        {

            e.printStackTrace();

        }

    }

}

Client

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Socket;

public class VD3Client {

    public static void main (String [] args){

        try{

            Socket socket = new Socket("localhost",12345);

            System.out.println("Connected to server: " + socket);

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

            out.println("Request from client");

             BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

             String response = in.readLine();

             System.out.println("Received response from server: " + response);

             socket.close();

             System.out.println("Disconnected from server: " + socket);

        } catch (IOException e)

        {

            e.printStackTrace();

        }

    }

}

A screenshot of a computer screen

Description automatically generated

VD4

Server

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.concurrent.Callable;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.util.concurrent.Future;

public class VD4Server {

    public static void main (String [] args){

        ExecutorService executorService = Executors.newFixedThreadPool(10);

        try{

            ServerSocket serverSocket = new ServerSocket(12345);

            System.out.println("Server started and listening on port 12345.....");

            while (true) {

                Socket clientSocket = serverSocket.accept();

                System.out.println("Client connected: " + clientSocket);

                Future<String> result = executorService.submit(new ClientHandler(clientSocket));

                String response = result.get();

                System.out.println("Received response from client: " + response);

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

                out.println("Response from server: " + response);

                clientSocket.close();

                System.out.println("Client disconnected: " + clientSocket);

            }

        } catch (IOException e)

        {

            e.printStackTrace();

        } catch (Exception e){

            e.printStackTrace();

        } finally{

            executorService.shutdown();

        }

    }

}

class ClientHandler implements Callable<String>{

    private Socket clientSocket;

    public ClientHandler (Socket clientSocket){

        this.clientSocket = clientSocket;

    }

    @Override

    public String call() throws Exception{

        BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

        String request = in.readLine();

        System.out.println("Received request from client: " + request);

        return "Processed: " + request;

    }

}

Client

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Socket;

public class VD4Client {

    public static void main (String [] args){

        try{

            Socket socket = new Socket("localhost",12345);

            System.out.println("Connected to server: " + socket);

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

            out.println("Request from client");

            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

            String response = in.readLine();

            System.out.println("Received response from server: " + response);

            socket.close();

            System.out.println("Disconnected from server: " + response);

            socket.close();

            System.out.println("Disconnected from server: " + socket);

        }catch(IOException e)

        {

            e.printStackTrace();

        }

    }

}

A screenshot of a computer screen

Description automatically generated

Bai1

Server

import java.io.\*;

import java.net.\*;

public class Bai1Server {

    public static void main(String[] args) {

        try (ServerSocket serverSocket = new ServerSocket(5000)) {

            System.out.println("Server is listening on port 5000");

            while (true) {

                Socket socket = serverSocket.accept();

                System.out.println("New client connected");

                new ServerThread(socket).start();

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

class ServerThread extends Thread {

    private Socket socket;

    public ServerThread(Socket socket) {

        this.socket = socket;

    }

    public void run() {

        try (BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

             PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

            String input;

            while ((input = in.readLine()) != null) {

                try {

                    int number = Integer.parseInt(input);

                    int square = number \* number;

                    System.out.println("Received number: " + number);

                    System.out.println("Square of the number: " + square);

                    out.println(square);

                } catch (NumberFormatException e) {

                    out.println("Invalid number");

                }

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

Client

import java.io.\*;

import java.net.\*;

public class Bai1Client {

    public static void main(String[] args) {

        try (Socket socket = new Socket("localhost", 5000);

             BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

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

             BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

            System.out.println("Connected to the server");

            String userInput;

            while (true) {

                System.out.print("Enter a number: ");

                userInput = reader.readLine();

                if ("exit".equalsIgnoreCase(userInput)) {

                    break;

                }

                out.println(userInput);

                String response = in.readLine();

                System.out.println("Square: " + response);

            }

        } catch (UnknownHostException e) {

            System.err.println("Don't know about host");

            e.printStackTrace();

        } catch (IOException e) {

            System.err.println("Couldn't get I/O for the connection");

            e.printStackTrace();

        }

    }

}

A screenshot of a computer screen

Description automatically generated

Bai2

Server

import java.io.\*;

import java.net.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Bai2Server {

    public static void main(String[] args) {

        try (ServerSocket serverSocket = new ServerSocket(5000)) {

            System.out.println("Server is listening on port 5000");

            while (true) {

                Socket socket = serverSocket.accept();

                System.out.println("New client connected");

                new ServerThread(socket).start();

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

class ServerThread extends Thread {

    private Socket socket;

    public ServerThread(Socket socket) {

        this.socket = socket;

    }

    public void run() {

        try (PrintWriter out = new PrintWriter(socket.getOutputStream(), true)) {

            SimpleDateFormat formatter = new SimpleDateFormat("HH:mm:ss");

            while (true) {

                String currentTime = formatter.format(new Date());

                System.out.println("Sending time to client: " + currentTime);

                out.println(currentTime);

                Thread.sleep(1000);

            }

        } catch (IOException | InterruptedException e) {

            e.printStackTrace();

        }

    }

}

Client

import java.io.\*;

import java.net.\*;

public class Bai2Client {

    public static void main(String[] args) {

        try (Socket socket = new Socket("localhost", 5000);

             BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

            System.out.println("Connected to the server");

            String serverTime;

            while ((serverTime = in.readLine()) != null) {

                System.out.println("Current Time: " + serverTime);

            }

        } catch (UnknownHostException e) {

            System.err.println("Don't know about host");

            e.printStackTrace();

        } catch (IOException e) {

            System.err.println("Couldn't get I/O for the connection");

            e.printStackTrace();

        }

    }

}

A screenshot of a computer

Description automatically generated

Bai3

Server

import java.io.\*;

import java.net.\*;

import java.sql.\*;

import java.text.DecimalFormat;

public class Bai3Server {

    // Sử dụng Integrated Security để kết nối

    private static final String DB\_URL = "jdbc:sqlserver://localhost:1433;databaseName=CreditCardDB;integratedSecurity=true;";

    public static void main(String[] args) {

        try (ServerSocket serverSocket = new ServerSocket(5000)) {

            System.out.println("Server đang lắng nghe trên cổng 5000");

            while (true) {

                Socket socket = serverSocket.accept();

                System.out.println("Client mới kết nối");

                new ServerThread(socket).start();

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

    static class ServerThread extends Thread {

        private Socket socket;

        public ServerThread(Socket socket) {

            this.socket = socket;

        }

        public void run() {

            try (BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

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

                 Connection connection = DriverManager.getConnection(DB\_URL)) {

                String cardNumber = in.readLine();

                String code = in.readLine();

                if (!isValidCardNumber(cardNumber) || !isValidCode(code)) {

                    out.println("Định dạng số thẻ hoặc mã không hợp lệ");

                    return;

                }

                String query = "SELECT card\_name, country, balance FROM credit\_card WHERE card\_number = ? AND code = ?";

                try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

                    preparedStatement.setString(1, cardNumber);

                    preparedStatement.setString(2, code);

                    ResultSet resultSet = preparedStatement.executeQuery();

                    if (resultSet.next()) {

                        String cardName = resultSet.getString("card\_name");

                        String country = resultSet.getString("country");

                        double balance = resultSet.getDouble("balance");

                        DecimalFormat df = new DecimalFormat("#,###.##");

                        String formattedBalance = df.format(balance) + " VND";

                        out.println("Tên thẻ: " + cardName);

                        out.println("Quốc gia: " + country);

                        out.println("Số dư: " + formattedBalance);

                    } else {

                        out.println("Không tìm thấy thẻ hoặc mã không hợp lệ");

                    }

                }

            } catch (IOException | SQLException e) {

                e.printStackTrace();

            }

        }

        private boolean isValidCardNumber(String cardNumber) {

            return cardNumber.matches("\\d{16}");

        }

        private boolean isValidCode(String code) {

            return code.length() <= 32;

        }

    }

}

Client

import java.io.\*;

import java.net.\*;

public class Bai3Client {

    public static void main(String[] args) {

        try (Socket socket = new Socket("localhost", 5000);

             BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

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

             BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

            System.out.println("Đã kết nối đến server");

            System.out.print("Nhập số thẻ: ");

            String cardNumber = reader.readLine();

            System.out.print("Nhập mã: ");

            String code = reader.readLine();

            out.println(cardNumber);

            out.println(code);

            String response;

            while ((response = in.readLine()) != null) {

                System.out.println(response);

            }

        } catch (UnknownHostException e) {

            System.err.println("Không biết về host");

            e.printStackTrace();

        } catch (IOException e) {

            System.err.println("Không thể lấy I/O cho kết nối");

            e.printStackTrace();

        }

    }

}

A screenshot of a computer program

Description automatically generated