

**ĐỀ THI VÀ BÀI LÀM**

Tên học phần: Lập Trình Mạng

Mã học phần: Hình thức thi: *Tự luận có giám sát*

Thời gian làm bài: 60 phút (*không kể thời gian phát đề và nộp bài*)

Được sử dụng tài liệu khi làm bài. Không chia sẻ bài cho nhau, nếu phát hiện sẽ chia đều số điểm.

**Họ tên:**... Vũ Thị Nga.....**Lớp:**.....19TCLC-DT5 .....**MSSV:**.....102190226 .....

Sinh viên làm bài trực tiếp trên tệp này, lưu tệp với định dạng MSSV\_HọTên.pdf và nộp bài thông qua MSTeam:

**Câu 1** (3 điểm): Hãy viết chương trình theo giao thức TCP với các chức năng sau:

a) Server:

- a. Lắng nghe và chấp nhận kết nối từ các client. Ứng với mỗi kết nối tạo 1 luồng xử lý sau:
  - i. Nhận chuỗi ký tự từ client gửi tới.
  - ii. Kiểm tra chuỗi đó có phải theo định dạng ngày tháng năm (DD/MM/YYYY) hay không. Nếu phải thì trả về cho client ngày tháng năm theo định dạng trên của 7 ngày sau, ngược lại thì trả về ngày của hệ thống. Ví dụ: Chuỗi nhận là “25/11/2021” thì trả về cho client chuỗi “02/12/2021”.
  - iii. Đóng kết nối

b) Client:

- a. Kết nối tới server
- b. Gửi chuỗi bất kỳ
- c. Nhận kết quả từ server gửi về và hiển thị.
- d. Đóng kết nối

# Trả lời:

**Dán code server vào bên dưới**

**ppackage** TCP.bai2;

```
import java.io.DataOutputStream;
import java.io.DataInputStream;
import java.net.*;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Server {
    public static void main(String[] args) throws Exception
    {
        ServerSocket server = new ServerSocket(7788);
        System.out.println("Server is started");
        while(true)
        {
```

```

        Socket socket = server.accept();
        DataOutputStream dos = new
DataOutputStream(socket.getOutputStream());
        DataInputStream din = new
DataInputStream(socket.getInputStream());
        String st = din.readUTF();
        try{
            SimpleDateFormat sdf = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            Date date = new
SimpleDateFormat("dd/MM/yyyy").parse(st);
            date.toString();
            dos.writeUTF("Server tra lai ngay gio = " +
date);
        }catch (Exception e){
            String time = new Date().toString();
            dos.writeUTF("Server tra lai ngay gio = " +
time);
        }

        socket.close();
    }
}
}

```

### **Dán code client vào bên dưới**

```

package TCP.bai2;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import java.util.Scanner;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JTextArea;
import javax.swing.JButton;

public class ClientTCP extends JFrame {

```

```

private JPanel contentPane;
private JTextField txtNhap;
private JTextField txtKq;

/**
 * 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();
            }
        }
    });
}

/**
 * Create the frame.
 */
public ClientTCP() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 431, 309);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    txtNhap = new JTextField();
    txtNhap.setBounds(82, 65, 239, 26);
    contentPane.add(txtNhap);
    txtNhap.setColumns(10);

    JLabel lblNewLabel = new JLabel("Nh\u1EADp chu\u1ED7i");
    lblNewLabel.setBounds(82, 40, 108, 14);
    contentPane.add(lblNewLabel);

    JLabel lblKtQu = new JLabel("K\u1EBFt qu\u1EA3");
    lblKtQu.setBounds(82, 161, 108, 14);
    contentPane.add(lblKtQu);

    JButton btnGui = new JButton("G\u1EEDi");
    btnGui.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent actionEvent) {
            try {
                String st = Ketqua();
                txtKq.setText(st);
            } catch (Exception e1) {

```

```

        e1.printStackTrace();
    }
}

});
btnGui.setBounds(162, 116, 89, 23);
contentPane.add(btnGui);

txtKq = new JTextField();
txtKq.setColumns(10);
txtKq.setBounds(82, 186, 239, 26);
contentPane.add(txtKq);
}
public String Ketqua() throws Exception
{
    Socket socket = new Socket("localhost",7889);
    DataInputStream din = new DataInputStream(socket.getInputStream());
    DataOutputStream dos = new DataOutputStream(socket.getOutputStream());
    String st = txtNhap.getText();
    System.out.print(st);
    dos.writeUTF(st);
    dos.flush();
    String st1 = din.readUTF();
    System.out.print(st1);
    return st1;
}
}

```

### **Dán kết quả thực thi vào bên dưới**

**Câu 2** (2 điểm): Hãy cho biết nếu chuyển chương trình ở câu 1 qua giao thức UDP thì các công việc cần làm gồm những gì? (Không cần code, chỉ cần mô tả)

Ở phía Server

Khi tạo socket sử dụng DatagramSocket.

Tạo biến receiveData kiểu byte[] để nhận dữ liệu từ gói tin đến , tạo biến DatagramPacket  
receivePacket = new DatagramPacket(receiveData, receiveData.length) để nhận gói tin từ socket.

Nhận gói tin từ phương thức receive() , chuyển dữ liệu kiểu byte của gói tin thành kiểu String.

Để gửi gói tin về lại client thì trước hết phải lấy địa chỉ IP và port của bên gửi.

Sau đó chuyển gói tin cần gửi sang kiểu byte[] và sử dụng phương thức send() để gửi gói tin đi.

Ở phía Client

Tạo socket để kết nối với Server

Tạo địa chỉ IP hoặc tên localhost

Để nhận và gửi gói tin thì cũng thực hiện như ở Server.

**Câu 3** (5 điểm): Trong phần bài tập JSP/Servlet đã nộp hãy trả lời các câu hỏi sau:

- a) Cho biết tên của các thành viên trong nhóm, kể cả bản thân

**Nguyễn Khánh Huy**  
**Đoàn Thị Diệu Hằng**  
**Vũ Thị Nga**

- b) Mô tả chức năng chính mà bản thân đã đóng góp vào trong chương trình.

**Giao diện của chương trình.**  
**Phản hướng dẫn cài đặt tomcat,xampp trong báo cáo.**

- c) *Hãy mô tả cách thức để có thể xử lý một request với thời gian lớn mà không bị “request time out”. Hãy trích tối đa 10 dòng code trong bài tập đã làm thể hiện điều đó! (có thể thêm comment nếu cần, nhưng không được trích quá 10 dòng code)*

**GIẢNG VIÊN BIÊN SOẠN ĐỀ THI**

Đà Nẵng, ngày 29 tháng 11 năm 2021  
**TRƯỞNG BỘ MÔN**  
(đã duyệt)