|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA  **KHOA Công Nghệ Thông Tin**  BỘ MÔN: MẠNG & TT |

**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:**Thân Nguyên Minh Quân **Lớp**:20TCLC-DT3 **MSSV**:102200148

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

1. Server:
   1. Lắng nghe và chấp nhận kết nối từ các client. Ứng với mỗi kết nối tạo 1 luồng xử lý sau:
      1. Nhận chuỗi ký tự từ client gửi tới.
      2. Kiểm tra chuỗi đó có phải theo định dạng ngày tháng năm (DD/MM/YYYY) hay không. Nếu phải thì kiểm tra ngày, tháng và năm của chuỗi đó có phải là số nguyên tố hay không. Nếu đúng thì trả về cho client chuỗi “Day la 1 ngay tuyet voi”. Ví dụ: Chuỗi nhận là “23/11/2027” thì trả về cho client chuỗi “Day la 1 ngay tuyet voi”.
      3. Đóng kết nối
2. Client:
   1. Kết nối tới server
   2. Gửi chuỗi bất kỳ
   3. Nhận kết quả từ server gửi về và hiển thị.
   4. Đóng kết nối

|  |
| --- |
| **# Trả lời:**  **Dán code server vào bên dưới**  package Cau1;  import java.io.DataInputStream;  import java.io.DataOutputStream;  import java.io.IOException;  import java.net.ServerSocket;  import java.net.Socket;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class server {  public static void main(String[] args) {  new server();  }  public server() {  try {  ServerSocket server = new ServerSocket(5000);  while (true) {  Socket s = server.accept();  Xuly x = new Xuly(s);  x.start();  }  } catch (Exception e) {  System.out.println(e);  }  }  }  class Xuly extends Thread {  Socket soc;  public Xuly(Socket s) {  soc = s;  }  private static Boolean DateValidator(String date) {  SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");  // To make strict date format validation  formatter.setLenient(false);  try {  formatter.parse(date);  return true;  } catch (Exception e) {  // Handle exception  System.out.println(e.getMessage());  return false;  }  }  public static boolean isPrimeNumber(int n) {  if (n < 2) {  return false;  }  int squareRoot = (int) Math.sqrt(n);  for (int i = 2; i <= squareRoot; i++) {  if (n % i == 0) {  return false;  }  }  return true;  }  private static String func(String date) {  if (DateValidator(date)) {  String[] arr = date.split("/");  for (int i = 0; i < arr.length; i++) {  int x = Integer.parseInt(arr[i]);  if (!isPrimeNumber(x)) {  return "ngày, tháng và năm của chuỗi đó khong phải là số nguyên tố ";  }  }  return "Day la 1 ngay tuyet voi";  } else {  return "Nhap sai(DD/MM/YYYY)";  }  }  @Override  public void run() {  try {  DataOutputStream dos = new DataOutputStream(soc.getOutputStream());  DataInputStream dis = new DataInputStream(soc.getInputStream());  while (true) {  String s = dis.readUTF();  String re = func(s);  System.out.println(re);  dos.writeUTF(re);  }  } catch (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  }  **Dán code client vào bên dưới**  package Cau1;  import java.awt.EventQueue;  import javax.swing.JFrame;  import javax.swing.JPanel;  import javax.swing.border.EmptyBorder;  import javax.swing.JTextField;  import java.awt.Font;  import javax.swing.JLabel;  import javax.swing.JButton;  import java.awt.event.ActionListener;  import java.io.DataInputStream;  import java.io.DataOutputStream;  import java.io.IOException;  import java.net.Socket;  import java.awt.event.ActionEvent;  public class client extends JFrame implements Runnable{  private JPanel contentPane;  private JTextField txtInput;  JLabel lblOutput;  DataInputStream dis;  DataOutputStream dos;  /\*\*  \* Launch the application.  \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  client frame = new client();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\*  \* Create the frame.  \*/  public client() {    try {  Socket soc = new Socket("localhost", 5000);  dis = new DataInputStream(soc.getInputStream());  dos = new DataOutputStream(soc.getOutputStream());  new Thread(this).start();  } catch (Exception e) {  }    setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 482, 282);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    txtInput = new JTextField();  txtInput.setFont(new Font("Tahoma", Font.PLAIN, 13));  txtInput.setBounds(191, 50, 166, 19);  contentPane.add(txtInput);  txtInput.setColumns(10);    lblOutput = new JLabel("");  lblOutput.setFont(new Font("Tahoma", Font.PLAIN, 13));  lblOutput.setBounds(191, 120, 220, 13);  contentPane.add(lblOutput);    JButton btnOK = new JButton("OK");  btnOK.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  try {  dos.writeUTF(txtInput.getText());  } catch (IOException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }  }  });  btnOK.setBounds(173, 195, 85, 21);  contentPane.add(btnOK);    JLabel lblNewLabel = new JLabel("Nhập ...");  lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 13));  lblNewLabel.setBounds(73, 55, 45, 13);  contentPane.add(lblNewLabel);    JLabel lblOutput\_1 = new JLabel("Kết quả");  lblOutput\_1.setFont(new Font("Tahoma", Font.PLAIN, 13));  lblOutput\_1.setBounds(73, 120, 45, 13);  contentPane.add(lblOutput\_1);  }  @Override  public void run() {  while(true) {  try {  String s = dis.readUTF();  System.out.println(s);  lblOutput.setText(s);  } catch (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }    }  }  }  **Dán kết quả thực thi vào bên dưới** |

***Câu 2*** (*3 điểm*): Hãy chuyển chương trình ở câu 1 qua giao thức UDP.

|  |
| --- |
| **# Trả lời:**  **Dán code server vào bên dưới**  package Cau2;  import java.io.DataInputStream;  import java.io.DataOutputStream;  import java.io.IOException;  import java.net.DatagramPacket;  import java.net.DatagramSocket;  import java.net.ServerSocket;  import java.net.Socket;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class server {  public static void main(String[] args) {  new server();  }  public server() {  try {  DatagramSocket TT = new DatagramSocket(5000);  while (true) {  DatagramPacket lt = new DatagramPacket(new byte[100], 100);  TT.receive(lt);  String s = func(new String(lt.getData()).substring(0,lt.getLength()));    DatagramPacket lt2 = new DatagramPacket(s.getBytes(), s.length(),lt.getAddress(),lt.getPort());  TT.send((lt2));  }  } catch (Exception e) {  System.out.println(e);  }  }    private static Boolean DateValidator(String date) {  SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");  // To make strict date format validation  formatter.setLenient(false);  try {  formatter.parse(date);  return true;  } catch (Exception e) {  // Handle exception  System.out.println(e.getMessage());  return false;  }  }  //Check snt for daymonthyear  public static boolean isPrimeNumber(int n) {  if (n < 2) {  return false;  }  int squareRoot = (int) Math.sqrt(n);  for (int i = 2; i <= squareRoot; i++) {  if (n % i == 0) {  return false;  }  }  return true;  }  private static String func(String date) {  if (DateValidator(date)) {  String[] arr = date.split("/");  for (int i = 0; i < arr.length; i++) {  int x = Integer.parseInt(arr[i]);  if (!isPrimeNumber(x)) {  return "Ngày, tháng và năm của chuỗi đó không phải là số nguyên tố ";  }  }  return "Day la 1 ngay tuyet voi";  } else {  return "Nhap sai (DD/MM/YYYY)";  }  }  }  **Dán code client vào bên dưới**  package Cau2;  import java.awt.EventQueue;  import javax.swing.JFrame;  import javax.swing.JPanel;  import javax.swing.border.EmptyBorder;  import javax.swing.JTextField;  import java.awt.Font;  import javax.swing.JLabel;  import javax.swing.JButton;  import java.awt.event.ActionListener;  import java.io.DataInputStream;  import java.io.DataOutputStream;  import java.io.IOException;  import java.net.DatagramPacket;  import java.net.DatagramSocket;  import java.net.InetAddress;  import java.net.Socket;  import java.net.SocketException;  import java.awt.event.ActionEvent;  public class client extends JFrame implements Runnable{  private JPanel contentPane;  private JTextField txtInput;  JLabel lblOutput;  DatagramSocket TT;  /\*\*  \* Launch the application.  \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  client frame = new client();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\*  \* Create the frame.  \*/  public client() {    try {  TT = new DatagramSocket();  } catch (SocketException e2) {  e2.printStackTrace();  }    new Thread(this).start();  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 482, 282);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    txtInput = new JTextField();  txtInput.setFont(new Font("Tahoma", Font.PLAIN, 13));  txtInput.setBounds(191, 50, 166, 19);  contentPane.add(txtInput);  txtInput.setColumns(10);    lblOutput = new JLabel("");  lblOutput.setFont(new Font("Tahoma", Font.PLAIN, 13));  lblOutput.setBounds(191, 120, 229, 13);  contentPane.add(lblOutput);    JButton btnOK = new JButton("OK");  btnOK.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  try {  String s = txtInput.getText();  DatagramPacket lt1 = new DatagramPacket(s.getBytes(), s.length(), InetAddress.getLocalHost(),  5000);  TT.send(lt1);  } catch (IOException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }  }  });  btnOK.setBounds(173, 195, 85, 21);  contentPane.add(btnOK);    JLabel lblNewLabel = new JLabel("Nhập Chuỗi");  lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 13));  lblNewLabel.setBounds(80, 60, 45, 13);  contentPane.add(lblNewLabel);    JLabel lblOutput\_1 = new JLabel("Kết Quả");  lblOutput\_1.setFont(new Font("Tahoma", Font.PLAIN, 13));  lblOutput\_1.setBounds(80, 130, 45, 13);  contentPane.add(lblOutput\_1);  }  public void run() {  while(true) {  try {  DatagramPacket lt = new DatagramPacket(new byte[100], 100);  TT.receive(lt);  lblOutput.setText(new String(lt.getData()).substring(0, lt.getLength()));  } catch (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }    }  }  }  **Dán kết quả thực thi vào bên dưới** |

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

1. Cho biết tên của các thành viên trong nhóm, kể cả bản thân

|  |
| --- |
| **# Trả lời:** Thân Nguyên Minh Quân  Nguyễn Văn Cường  Nguyễn Minh Quốc |

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

|  |
| --- |
| **# Trả lời:** Chức năng Login,Logout .(Đăng nhập đăng xuất trang web)  -Chức năng login thì em đăng nhập vào sử dụng session nhận diện (chỉ có 1 role là admin)  -Logout thì em dùng hàm trả về lại trang đăng nhập là Login.jsp  Chức năng Read (List các sản phẩm từ database)  -Hiển thị các sản phẩm từ database xuất ra bảng ở website (Cùng 1 form với các chức năng khác ngoại trừ Update)  Chức năng Delete(Xóa sản phẩm đã chọn) Chỉ cần click vào button delete bên cạnh mỗi sản phẩm , không hiển thị message box .  \*Luồng đi của Delete  Xóa thẳng và trả về bảng list sản phẩm là vị trí của table trong database . |

Đà Nẵng, ngày 12 tháng 12 năm 2022

|  |  |
| --- | --- |
| **GIẢNG VIÊN BIÊN SOẠN ĐỀ THI** | **TRƯỞNG BỘ MÔN** |
|  | (đã duyệt) |