Ngày báo cáo: 1/11/2020

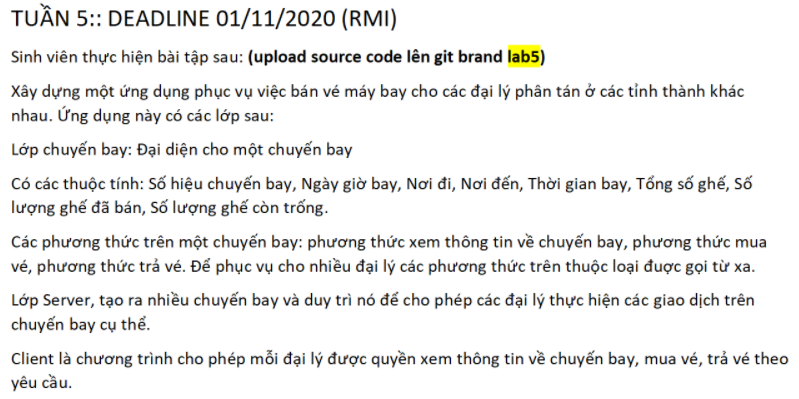
Module báo cáo: LAB 5

Mã sinh viên: 18065791

Họ và tên: Phùng Thị Hằng

**NỘI DUNG**

**LAB 5:**



**Code:**

Server:

Package tuan5;

import java.net.MalformedURLException;

import java.rmi.AlreadyBoundException;

import java.rmi.Naming;

import java.rmi.RemoteException;

import java.rmi.registry.LocateRegistry;

import IFlight;

import FlightImpl;

public class Server {

    public static void main(String args[]) {

        try {

            LocateRegistry.createRegistry(2007);

            // Sign in rmiregistry

            IFlight x = new FlightImpl();

            Naming.bind("rmi://localhost:2007/in4RMI", x );

            System.out.println("<<<< SERVER ACTIVATED >>>>");

        } catch (RemoteException e) {

            e.printStackTrace();

        } catch (AlreadyBoundException e) {

            e.printStackTrace();

        } catch (MalformedURLException e) {

            e.printStackTrace();

        }

    }

}

Client:

Package tuan5;

import java.net.MalformedURLException;

import java.rmi.Naming;

import java.rmi.NotBoundException;

import java.rmi.RemoteException;

import java.util.Scanner;

import IFlight;

public class Client {

    public static void main(String args[]) {

        try {

            IFlight iFlight = (IFlight) Naming.lookup("rmi://localhost:2007/in4RMI");

            System.out.println("Connect Success!!!");

            System.out.println(iFlight.Display(0));

            String[] id = iFlight.DataId();

            for (int i = 0; i < id.length; i++) {

                System.out.println((i+1)+">\tIdFlight: "+id[i]+"\t\tFrom: "+iFlight.getFlight(id[i]).getFrom()+"\t\tTo: "+iFlight.getFlight(id[i]).getTo()+"\t\tDate: "+iFlight.getFlight(id[i]).getDate());

            }

            while(true){

                System.out.print(">>>");

                Scanner scan= new Scanner(System.in);

                String text = scan.nextLine();

                //

                if(text.matches("[0-9]+")) { // [0-9]+ or \\d+

                    int x = Integer.parseInt(text);

                    //

                    if(x <= id.length && x > 0) {

                        System.out.println(iFlight.Display(1));

                        System.out.println("ID :\t"+iFlight.getFlight(id[x-1]).getId());

                        System.out.println("From : "+iFlight.getFlight(id[x-1]).getFrom()+"\t====>\tTo :"+iFlight.getFlight(id[x-1]).getTo());

                        System.out.println("Date :\t"+iFlight.getFlight(id[x-1]).getDate());

                        System.out.println("Total number of tickets :\t"+iFlight.getFlight(id[x-1]).getTotal());

                        System.out.println("Tickets sold : "+iFlight.getFlight(id[x-1]).getOdered()+"\t<----->\t Tickets not yet sold :"+(iFlight.getFlight(id[x-1]).getTotal()-iFlight.getFlight(id[x-1]).getOdered()));

                        //

                        do {

                            int sold = iFlight.getFlight(id[x-1]).getOdered();

                            int total = iFlight.getFlight(id[x-1]).getTotal();

                            System.out.print(">>>");

                            Scanner scan1 = new Scanner(System.in);

                            String txt = scan1.nextLine();

                            if(txt.matches("[0-9]+")) {

                                int x1 = Integer.parseInt(txt);

                                if(x1 == 1) {

                                    //

                                    if(sold >= total) {

                                        System.out.println(" Not enough ticket ");

                                        scan1.nextLine();

                                    }else {

                                        while(true) {

                                            System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                                            System.out.print(" Number of tickets: >>>");

                                            scan1 = new Scanner(System.in);

                                            String od = scan1.nextLine();

                                            //

                                            if(od.matches("[0-9]+")){

                                                int x2 = Integer.parseInt(od);

                                                if(sold+x2 < total) {

                                                    System.out.print("Username : ");

                                                    String u = scan1.nextLine();

                                                    System.out.print("Password : ");

                                                    String p = scan1.nextLine();

                                                    System.out.println(iFlight.Order(id[x-1], x2,u,p));

                                                    scan1.nextLine();

                                                    break;

                                                }else System.out.println("number is large more total tickets");

                                            }else if(od.equals("cancel")) { break; }

                                            else System.out.println("Input is not true");

                                            //

                                        }

                                    }

                                    //

                                    System.out.println(iFlight.Display(0));

                                    for (int i = 0; i < id.length; i++) {

                                        System.out.println((i+1)+">\tIdFlight: "+id[i]+"\t\tFrom: "+iFlight.getFlight(id[i]).getFrom()+"\t\tTo: "+iFlight.getFlight(id[i]).getTo()+"\t\tDate: "+iFlight.getFlight(id[i]).getDate());

                                    }

                                    break;

                                    }else

                                    if(x1 == 2) {

                                    System.out.println(iFlight.Display(0));

                                    for (int i = 0; i < id.length; i++) {

                                        System.out.println((i+1)+">\tIdFlight: "+id[i]+"\t\tFrom: "+iFlight.getFlight(id[i]).getFrom()+"\t\tTo: "+iFlight.getFlight(id[i]).getTo()+"\t\tDate: "+iFlight.getFlight(id[i]).getDate());

                                    }

                                    break;

                                }else System.out.println("Input data errol!!!");

                            }else System.out.println("Input data errol!!!");

                        }while(true);

                        //

                    }else System.out.println("Input number not true !!!");

                    //

                }else if(text.equals("exit")) break;

                        else System.out.println("Input data errol!!! Input must is number");

                //

            }

        } catch (NotBoundException e) {

            e.printStackTrace();

        } catch (MalformedURLException e) {

            e.printStackTrace();

        } catch (RemoteException e) {

            e.printStackTrace();

        }

    }

}

Interface:

Package tuan5;

import java.rmi.Remote;

import java.rmi.RemoteException;

import Flight;

public interface IFlight extends Remote{

    public Flight getFlight(String x) throws RemoteException;

    public String Display(int x) throws RemoteException;

    public String[] DataId() throws RemoteException;

    public String Order(String x,int y,String user,String password) throws RemoteException;

}

Object

Package tuan5;

import java.io.Serializable;

public class Flight implements Serializable{

    private static final long serialVersionUID = 1L;

    private int Total,Odered;

    private String Flytime,Date,Id,From,To;

    public String getDate() { return Date; }

    public void setDate(String x) { this.Date = x; }

    public String getFlyTime() { return Flytime; }

    public void setFlyTime(String x) { this.Flytime = x; }

    public int getTotal() { return Total; }

    public void setTotal(int x) { this.Total = x; }

    public int getOdered() { return Odered; }

    public void setOdered(int x) { this.Odered = x; }

    public String getId() { return Id; }

    public void setId(String x) { this.Id = x; }

    public String getFrom() { return From; }

    public void setFrom(String x) { this.From = x; }

    public String getTo() { return To; }

    public void setTo(String x) { this.To = x; }

}

Thuc thi

Package tuan5;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.util.Scanner;

import Flight;

import IFlight;

public class FlightImpl extends UnicastRemoteObject implements IFlight{

    /\*\*

     \*

     \*/

    private static final long serialVersionUID = 1L;

    public FlightImpl() throws RemoteException {

    }

    //--------------Data--------------

    public Flight[] Data(){

        File f = new File("F:\\flight.txt");

        Scanner c = null;

        try {

            c = new Scanner(f,"UTF-8");

        } catch (FileNotFoundException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

        Flight data[] = new Flight[100];

        int i = 0;

        while(c.hasNextLine()) {

            String line = c.nextLine();

            String[] parts = line.split("\t");

            data[i] = new Flight();

            data[i].setId(parts[0]);

            data[i].setDate(parts[1]);

            data[i].setFrom(parts[2]);

            data[i].setTo(parts[3]);

            data[i].setFlyTime(parts[4]);

            data[i].setTotal(Integer.parseInt(parts[5]));

            data[i].setOdered(Integer.parseInt(parts[6]));

            i++;

            }

        Flight Data[] = new Flight[i];

        for (int j = 0; j < Data.length; j++) {

            Data[j] = data[j];

        }

        return Data;

    }

    //------------getFlight = findId------------

    @Override

    public Flight getFlight(String x){

        // TODO Auto-generated method stub

        Flight[] data = Data();

        Flight DataFind = new Flight();

        for (int i = 0; i < data.length; i++) {

            if (data[i].getId().equals(x)) {

                DataFind = data[i];

                break;

            }

        }

        return DataFind;

    }

    //------------Id Flight----------------

    @Override

    public String[] DataId() throws RemoteException {

        // TODO Auto-generated method stub

        Flight[] data = Data();

        String[] id = new String[data.length];

        for (int i = 0; i < data.length; i++) {

            id[i] = new String();

            id[i] = data[i].getId();

        }

        return id;

    }

    //-----------Edit data--------------

    //------------Display Client--------------

    @Override

    public String Display(int x) throws RemoteException {

        // TODO Auto-generated method stub

        String s = new String();

        switch(x) {

        case 0:{

            s =".-----------------------------------------------------------------------------------------------."

                    +"\n"+"|                    AIRLINE HELLO !!!                   |"

                    +"\n"+"\*-----------------------------------------------------------------------------------------------\*"

                    +"\n"+" (\*)Choose the number corresponding to the flight\t"+"\t"+"(\*)Type 'Exit' if you want to leave"+"\n\n";

            break;

        }

        case 1:{

             s =".-----------------------------------------------------------------------------------------------."

                        +"\n"+"|                    Flight Details !!!                  |"

                        +"\n"+"\*-----------------------------------------------------------------------------------------------\*"

                        +"\n"+"\t\tFlight Details:\n\t\t(Choose 1 or 2) "+"1.Order Tickets\t\t\t"+"2.Cancel"+"\n";

             break;

        }

        case 2:{

             s =".-----------------------------------------------------------------------------------------------."

                        +"\n"+"|                    Order Ticket !!!                    |"

                        +"\n"+"\*-----------------------------------------------------------------------------------------------\*"

                        +"\n"+"\t\tTicket:\n\t\t"+"Write return to exit"+"\n\n";

             break;

        }

        default:{

            s = "Data is not true !!! plesase press again !!!\n\n";

            break;

        }

        }

        return s;

    }

    @Override

    public String Order(String x, int y,String user,String password) throws RemoteException {

        String s = null;

        // TODO Auto-generated method stub

        if(user.equals("tranmanhthin")&&password.equals("05112000")) {

            Flight[] data = Data();

            for (int i = 0; i < data.length; i++) {

                if(data[i].getId().equals(x)) {

                    //data[i].setOdered(data[i].getOdered()+y);

                    data[i].setOdered(data[i].getOdered()+y);

                }

            }

            try {

                FileWriter writer = new FileWriter("F:\\flight.txt");

                for (int i = 0; i < data.length; i++) {

                    writer.write(data[i].getId()+"\t"+data[i].getDate()+"\t"+data[i].getFrom()+"\t"+data[i].getTo()+"\t"+data[i].getFlyTime()+"\t"+data[i].getTotal()+"\t"+data[i].getOdered()+"\n");

                }

                writer.close();

                s = "SUCCESS";

            } catch (IOException e) {

                // TODO Auto-generated catch block

                e.printStackTrace();

            }

        }else{

            s = "UNSUCCESS";

        }

        return s;

    }

}

Source txt



**(\*) Result:**



