Input frame code:

import java.awt.Color;

import java.awt.Component;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.util.ArrayList;

import javax.swing.BorderFactory;

import javax.swing.BoxLayout;

import javax.swing.DefaultListModel;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JList;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class InputFrame extends JFrame {

    private JPanel panel;

    private JPanel diamoniPanel;

    private JPanel allInclusivePanel;

    private JPanel buttonPanel;

    private JLabel diamoniLabel;

    private JLabel hotelLabel;

    private JLabel allInclusiveLabel;

    private JList listView;

    private DefaultListModel model;

    private JTextField daysField;

    private JTextField mealsField;

    private JButton storeButton;

    private JButton calculateCostButton;

    private JTextField costField;

    private ArrayList<Hotel> hotels;

    public InputFrame() {

        panel = new JPanel();

        diamoniPanel = new JPanel();

        allInclusivePanel = new JPanel();

        buttonPanel = new JPanel();

        File file = new File("hotels.ser");

        try {

            FileInputStream fileIn = new FileInputStream(file);

            ObjectInputStream in = new ObjectInputStream(fileIn);

            hotels = (ArrayList<Hotel>) in.readObject();

            in.close();

            fileIn.close();

        } catch (FileNotFoundException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        } catch (IOException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        } catch (ClassNotFoundException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

        listView = new JList();

        model = new DefaultListModel();

        for(Hotel hotel: hotels)

            model.addElement(hotel.getName());

        listView.setModel(model);

        diamoniLabel = new JLabel("Stoixeia Diamonis");

        hotelLabel = new JLabel("Hotel");

        allInclusiveLabel = new JLabel("AllInclusive");

        daysField = new JTextField("Hmeres Diamonis");

        mealsField = new JTextField("Plithos Geumatwn (1,2,3)");

        costField = new JTextField("Synoliko Kostos");

        storeButton = new JButton("Apothikeusi Kratisis");

        calculateCostButton = new JButton("Ypologismos Kostous");

        diamoniPanel.setLayout(new BoxLayout(diamoniPanel, BoxLayout.Y\_AXIS));

        diamoniLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

        diamoniPanel.add(diamoniLabel);

        hotelLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

        diamoniPanel.add(hotelLabel);

        diamoniPanel.add(listView);

        diamoniPanel.add(daysField);

        diamoniPanel.setBorder(BorderFactory.createLineBorder(Color.black));

        allInclusivePanel.setLayout(new GridLayout(2,0));

        allInclusivePanel.add(allInclusiveLabel);

        allInclusivePanel.add(mealsField);

        allInclusivePanel.setBorder(BorderFactory.createLineBorder(Color.black));

        buttonPanel.setLayout(new GridLayout(2,0));

        buttonPanel.add(storeButton);

        buttonPanel.add(calculateCostButton);

        buttonPanel.setBorder(BorderFactory.createLineBorder(Color.black));

        panel.add(diamoniPanel);

        panel.add(allInclusivePanel);

        panel.add(buttonPanel);

        panel.add(costField);

        this.setContentPane(panel);

        ButtonListener listener = new ButtonListener();

        storeButton.addActionListener(listener);

        CostButtonListener costListener = new CostButtonListener();

        calculateCostButton.addActionListener(costListener);

        this.setVisible(true);

        this.setSize(200, 320);

        this.setLocation(200, 0);

        this.setTitle("Input");

        this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

    }

    class ButtonListener implements ActionListener {

        public void actionPerformed(ActionEvent e) {

            String selectedHotelName = (String)listView.getSelectedValue();

            Hotel selectedHotel = null;

            for(Hotel hotel: hotels)

                if(hotel.getName().equals(selectedHotelName))

                    selectedHotel = hotel;

            String daysAsText = daysField.getText();

            int days = Integer.parseInt(daysAsText);

            String mealsAsText = mealsField.getText();

            int meals = Integer.parseInt(mealsAsText);

            Reservation res;

            if(meals == 0) {

                //���������� ����� ��������

                res = new Reservation(days);

            }

            else {

                //���������� all inclusive

                res = new AllInclusive(days, meals);

            }

            selectedHotel.addReservation(res);

            System.out.println("Reservation created and added");

        }

    }

    class CostButtonListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            String selectedHotelName = (String)listView.getSelectedValue();

            Hotel selectedHotel = null;

            for(Hotel hotel: hotels)

                if(hotel.getName().equals(selectedHotelName))

                    selectedHotel = hotel;

            double totalCharge = selectedHotel.calculateTotalCharge()

String totalChargeText = Double.toString(totalCharge);

            costField.setText(totalChargeText);

        }

    }

}

**Serialization**

**import** java.io.FileInputStream;

**public** **class** SerializationDemo {

**private** ArrayList<Employee> employees = **new** ArrayList<Employee>();

**public** SerializationDemo() {

Company C1 = **new** Company("Coca Cola");

Company C2 = **new** Company("Google");

employees.add(**new** Employee("John", C1));

employees.add(**new** Employee("Bob", C2));

serializing();

ArrayList<Employee> employeesFromFile = deserializing();

**for**(Employee e: employeesFromFile) {

e.printInfo();

}

}

**public** **void** serializing() {

**try** {

FileOutputStream fileOut = **new** FileOutputStream("employee.ser");

ObjectOutputStream out = **new** ObjectOutputStream(fileOut);

out.writeObject(employees);

out.close();

fileOut.close();

}

**catch**(IOException i) {

i.printStackTrace();

}

**finally** {

System.***out***.println("Serialization Attempted...");

}

}

**public** ArrayList<Employee> deserializing() {

**try** {

FileInputStream fileIn = **new** FileInputStream("employee.ser");

ObjectInputStream in = **new** ObjectInputStream(fileIn);

ArrayList<Employee> employees = (ArrayList<Employee>) in.readObject();

in.close();

fileIn.close();

}

**catch**(IOException i) {

i.printStackTrace();

}

**catch**(ClassNotFoundException c) {

c.printStackTrace();

}

**finally** {

System.***out***.println("De-Serialization Attempted...");

**return** employees;

}

}}