

```

import java.util.Scanner;
import java.util.Vector;
import java.util.HashSet;
import java.util.Iterator;
import java.io.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;

class PhoneInfo implements Serializable
{
    String name;
    String phoneNumber;
    public PhoneInfo(String name, String num)
    {
        this.name = name;
        phoneNumber = num;
    }
    public void showPhoneInfo()
    {
        System.out.println("name: "+name);
        System.out.println("phone: "+phoneNumber);
    }
    public String toString()
    {
        return "name: "+name+'\n'+"phone: "+phoneNumber+'\n';
    }
    public int hashCode()
    {
        return name.hashCode();
    }

    public boolean equals(Object obj)
    {
        PhoneInfo cmp = (PhoneInfo)obj;
        if(name.compareTo(cmp.name) == 0)
            return true;
        else
            return false;
    }
}

class PhoneBookManager
{
    public final File dataFile = new File("PhoneBook.dat");
    HashSet<PhoneInfo> infoStorage = new HashSet<PhoneInfo>();

    static PhoneBookManager inst = null;
    public static PhoneBookManager createManagerInst()
    {
        if(inst == null)
            inst = new PhoneBookManager();
        return inst;
    }

    private PhoneBookManager()
    {
        readFromFile();
    }
    public String searchData(String name)
    {
        PhoneInfo info = search(name);
        if(info == null)
            return null;
        else
            return info.toString();
    }
    public boolean deleteData(String name)
    {
        Iterator<PhoneInfo> itr=infoStorage.iterator();
        while(itr.hasNext())
        {
            PhoneInfo curInfo=itr.next();
            if(name.compareTo(curInfo.name) == 0)
            {
                itr.remove();
                return true;
            }
        }
    }
}

```

```

        return false;
    }
    private PhoneInfo search(String name)
    {
        Iterator<PhoneInfo> itr = infoStorage.iterator();
        while(itr.hasNext())
        {
            PhoneInfo curInfo=itr.next();
            if(name.compareTo(curInfo.name) == 0)
                return curInfo;
        }
        return null;
    }

    public void storeToFile()
    {
        try
        {
            FileOutputStream file = new FileOutputStream(dataFile);
            ObjectOutputStream out = new ObjectOutputStream(file);
            Iterator<PhoneInfo> itr=infoStorage.iterator();
            while(itr.hasNext())
                out.writeObject(itr.next());
            out.close();
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
    }

    public void readFromFile()
    {
        if(dataFile.exists() == false)
            return;
        try
        {
            FileInputStream file = new FileInputStream(dataFile);
            ObjectInputStream in = new ObjectInputStream(file);
            while(true)
            {
                PhoneInfo info = (PhoneInfo)in.readObject();
                if(info == null)
                    break;
                infoStorage.add(info);
            }
            in.close();
        }
        catch(IOException e)
        {
            return;
        }
        catch(ClassNotFoundException e)
        {
            return;
        }
    }
}

class SearchEventHandler implements ActionListener
{
    JTextField searchField;
    JTextArea textArea;

    public SearchEventHandler(JTextField field, JTextArea area)
    {
        searchField=field;
        textArea=area;
    }

    public void actionPerformed(ActionEvent e)
    {
        String name = searchField.getText();
        PhoneBookManager manager=PhoneBookManager.createManagerInst();
        String srchResult = manager.searchData(name);
        if(srchResult == null)
        {
            textArea.append("Search Failed: info does not exist.\n");
        }
        else
        {

```

```

        textArea.append("Search Completed.\n");
        textArea.append(srchResult);
        textArea.append("\n");
    }
}

class AddEventHandler implements ActionListener
{
    JTextField name;
    JTextField phone;
    JTextArea text;
    Vector<String> inputList = new Vector<String>();
    boolean isAdded;

    PhoneInfo info;
    public AddEventHandler(JTextField nameField, JTextField phoneField, JTextArea textArea)
    {
        name = nameField;
        phone = phoneField;
        text = textArea;
    }
    public void actionPerformed(ActionEvent e)
    {
        PhoneBookManager manager = PhoneBookManager.createManagerInst();
        info = new PhoneInfo(name.getText(), phone.getText());
        isAdded = manager.infoStorage.add(info);
        if(isAdded)
        {
            text.append("Update Completed.\n");
        }
        else
        {
            text.append("Update Failed: info already exist.\n");
        }
    }
}

class DeleteEventHandler implements ActionListener
{
    JTextField delField;
    JTextArea textArea;

    public DeleteEventHandler(JTextField field, JTextArea area)
    {
        delField = field;
        textArea = area;
    }

    public void actionPerformed(ActionEvent e)
    {
        String name = delField.getText();
        PhoneBookManager manager = PhoneBookManager.createManagerInst();
        boolean isDeleted = manager.deleteData(name);
        if(isDeleted)
            textArea.append("Remove Completed.\n");
        else
            textArea.append("Remove Failed: info does not exist.\n");
    }
}

class MainFrame extends JFrame
{
    JTextField srchField = new JTextField(15);
    JButton srchBtn = new JButton("SEARCH");

    JButton addBtn = new JButton("ADD");
    JRadioButton rbtn1 = new JRadioButton("General");

    ButtonGroup buttonGroup = new ButtonGroup();

    JLabel nameLabel = new JLabel("NAME");
    JTextField nameField = new JTextField(15);
    JLabel phoneLabel = new JLabel("PHONE NUMBER");
    JTextField phoneField = new JTextField(15);
    JTextField delField = new JTextField(15);
    JButton delBtn = new JButton("DEL");
    JTextArea textArea = new JTextArea(10, 25);
    public MainFrame(String title)

```

```

{
    super(title);
    setBounds(100, 200, 330, 450);
    setSize(730,350);
    setLayout(new GridLayout(0,2,0,0));
    Border border = BorderFactory.createEtchedBorder()
Border srchBorder = BorderFactory.createTitledBorder(border, "Search");
    JPanel srchPanel = new JPanel();
    srchPanel.setBorder(srchBorder);
    srchPanel.setLayout(new FlowLayout());
    srchPanel.add(srchField);
    srchPanel.add(srchBtn);
    Border addBorder=BorderFactory.createTitledBorder(border, "Add");
    JPanel addPanel = new JPanel();
    addPanel.setBorder(addBorder);
    addPanel.setLayout(new FlowLayout());

    JPanel addInputPanel = new JPanel();
    addInputPanel.setLayout(new GridLayout(0,2,5,5));
    buttonGroup.add(rbtn1);
    addPanel.add(addBtn);
    addInputPanel.add(nameLabel);
    addInputPanel.add(nameField);
    addInputPanel.add(phoneLabel);
    addInputPanel.add(phoneField);
    rbtn1.setSelected(true);
    addPanel.add(addInputPanel);
Border delBorder = BorderFactory.createTitledBorder(border, "Delete");
    JPanel delPanel = new JPanel();
    delPanel.setBorder(delBorder);
    delPanel.setLayout(new FlowLayout());
    delPanel.add(delField);
    delPanel.add(delBtn);
    JScrollPane scrollTextArea = new JScrollPane(textArea);
    Border textBorder=BorderFactory.createTitledBorder(border, "Infomation Board");
    scrollTextArea.setBorder(textBorder);
    JPanel actionPanel = new JPanel();
    actionPanel.setLayout(new BorderLayout());
    actionPanel.add(srchPanel, BorderLayout.NORTH);
    actionPanel.add(addPanel, BorderLayout.CENTER);
    actionPanel.add(delPanel, BorderLayout.SOUTH);
add(actionPanel);
    add(scrollTextArea);
    srchBtn.addActionListener(new SearchEventHandler(srchField, textArea));
    addBtn.addActionListener(new AddEventHandler(nameField, phoneField, textArea));
    delBtn.addActionListener(new DeleteEventHandler(delField, textArea));

    setVisible(true);
    setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
}
}

class PhoneBook
{
    public static void main(String[] args)
    {
        PhoneBookManager manager = PhoneBookManager.createManagerInst();
        MainFrame winFrame = new MainFrame("Phone By Anjali");
    }
}

```

