

Module Eleven - Assignment Submission

The Assignment

Write a program that uses one of the Map classes to implement a contact list. Contact information should include first name, last name, a phone number and an email address. The list should be stored on your computer's file system and entries sorted alphabetically according to a person's last name. The program should prompt the user for the name of the file where the information is stored, and allow the user to add a contact from the list, delete a contact from the list, and display the entire contact list.

Design

This seems like a very straightforward program. We'll need some local struct to store the data as we input it, which will just have to be a class which is serializable so we can store it.

We're gonna use a Map to store the contacts. We can use a tree map with the key being the last name, so we can just have it automatically sorted by last name alphabetically.

The Map helps with lookup too—if we want to delete a contact, we can simply check for the last name and remove it directly. We don't need to worry about keeping track of the order manually, which makes things easier. The only thing to keep in mind is that the map uses the last name as the key, so if there are two people with the same last name, we'll have a problem. I assume unique last names though

To store it, we just serialize the map into a file, and load on the next start up

Implementation

```
package module11;

import java.io.*;
import java.util.*;

// Class which manages a contact list as a TreeMap
// Program takes in file name as input and allows user
// to add, delete, and display contacts
public class ContactList {
    // Contact class to store individual contact details
    static class Contact implements Serializable {
        String firstName;
        String lastName;
        String phoneNumber;
        String email;

        public Contact(String firstName, String lastName, String
phoneNumber, String email) {
            this.firstName = firstName;
```

```
        this.lastName = lastName;
        this.phoneNumber = phoneNumber;
        this.email = email;
    }

    @Override
    public String toString() {
        return firstName + " " + lastName + " | " + phoneNumber + " | "
+ email;
    }
}

// TreeMap to store contacts sorted by last name
private static Map<String, Contact> contactList = new TreeMap<>();

public static void main(String[] args) throws IOException {
    Scanner scanner = new Scanner(System.in);

    // Prompt user for the filename to store contacts
    System.out.print("Enter the name of the file" +
        " where contact list will be stored: ");
    String fileName = scanner.nextLine();
    loadContacts(fileName);

    while (true) {
        // Display options to the user
        System.out.println("\nChoose an option:");
        System.out.println("1. Add a contact");
        System.out.println("2. Delete a contact");
        System.out.println("3. Display all contacts");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        switch (choice) {
            case 1:
                addContact(scanner);
                break;
            case 2:
                deleteContact(scanner);
                break;
            case 3:
                displayContacts();
                break;
            case 4:
                saveContacts(fileName);
                System.out.println("Goodbye!");
                return;
            default:
                System.out.println("Invalid choice, please try
again.");
        }
    }
}
```

```
}

private static void loadContacts(String fileName) throws IOException {
    // Check if file exists and load data
    File file = new File(fileName);
    if (file.exists()) {
        try (ObjectInputStream ois =
            new ObjectInputStream(new FileInputStream(file))) {
            // Safely deserialize the Map object
            Object object = ois.readObject();
            if (object instanceof Map) {
                Map<?, ?> map = (Map<?, ?>) object;
                contactList = new TreeMap<>();
                for (Map.Entry<?, ?> entry : map.entrySet()) {
                    if (entry.getKey() instanceof String &&
                        entry.getValue() instanceof Contact) {
                        contactList.put((String) entry.getKey(),
                                      (Contact) entry.getValue());
                    }
                }
            } else {
                System.out.println("The data in the file" +
                                   " is not in the expected format.");
            }
        } catch (ClassNotFoundException | IOException e) {
            System.out.println("Error loading contacts: " +
                               e.getMessage());
        }
    }
}

private static void saveContacts(String fileName) throws IOException {
    // Save contact list to file
    try (ObjectOutputStream oos = new
        ObjectOutputStream(new FileOutputStream(fileName))) {
        oos.writeObject(contactList);
    }
}

private static void addContact(Scanner scanner) {
    // Prompt user for contact details
    System.out.print("Enter first name: ");
    String firstName = scanner.nextLine();
    System.out.print("Enter last name: ");
    String lastName = scanner.nextLine();
    System.out.print("Enter phone number: ");
    String phoneNumber = scanner.nextLine();
    System.out.print("Enter email address: ");
    String email = scanner.nextLine();

    // Create new contact and add to the contact list
    Contact newContact = new
        Contact(firstName, lastName, phoneNumber, email);
    contactList.put(lastName, newContact);
}
```

```

        System.out.println("Contact added successfully.");
    }

    private static void deleteContact(Scanner scanner) {
        // Ask user for the last name of the contact to delete
        System.out.print("Enter the last name of the contact to delete: ");
        String lastName = scanner.nextLine();

        if (contactList.containsKey(lastName)) {
            contactList.remove(lastName);
            System.out.println("Contact deleted successfully.");
        } else {
            System.out.println("No contact found with that last name.");
        }
    }

    private static void displayContacts() {
        // Display all contacts sorted by last name
        if (contactList.isEmpty()) {
            System.out.println("No contacts found.");
        } else {
            System.out.println("\nContact List:");
            for (Contact contact : contactList.values()) {
                System.out.println(contact);
            }
        }
    }
}

```

Output:

```

Script started on 2025-04-02 02:05:13-04:00 [TERM="xterm-256color"
TTY="/dev/pts/1" COLUMNS="263" LINES="17"]
arkosh@greyskull:
~/repos/601_201/module11$ java ContactList.java PrintData.java test.csv
57$ java ContactList.java
arkosh@greyskull:
Enter the name of the file where contact list will be stored: contactlist

Choose an option:
1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit
Enter your choice: 1
Enter first name: Test
Enter last name: Name
Enter phone number: 12312312345 ✓ ✓

```

Enter email address: dfsjoifadsoji@gmail.com
Contact added successfully.

Choose an option:

1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit

Enter your choice: 1

Enter first name: Mister

Enter last name: Nu✓ ✓imbus

Enter phone number: 8✓ ✓91191191111

Enter email address: police@!✓ ✓gmail.com

Contact added successfully.

Choose an option:

1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit

Enter your choice: 3

Contact List:

Test Name | 1231231234 | dfsjoifadsoji@gmail.com

Mister Nimbus | 91191191111 | police@gmail.com

Choose an option:

1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit

Enter your choice: 1

Enter first name: Alhpa✓ ✓✓ ✓✓ ✓pha

Enter last name: Beta

Enter phone number: 1234315223✓ ✓

Enter email address: nononoinoinionoi102@gmail.com

Contact added successfully.

Choose an option:

1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit

Enter your choice: 3

Contact List:

Alpha Beta | 123431522 | nononoinoinionoi102@gmail.com

Test Name | 1231231234 | dfsjoifadsoji@gmail.com

Mister Nimbus | 91191191111 | police@gmail.com

Choose an option:

1. Add a contact
2. Delete a contact
3. Display all contacts

```
4. Exit
Enter your choice: 2
Enter the last name of the contact to delete: Beta
Contact deleted successfully.

Choose an option:
1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit
Enter your choice: 4
Goodbye!
arkosh@greyskull:
~/repos/601_201/module11$ java ContactList.java
Enter the name of the file where contact list will be stored: contactlist

Choose an option:
1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit
Enter your choice: 3

Contact List:
Test Name | 1231231234 | dfsjoifadsoji@gmail.com
Mister Nimbus | 91191191111 | police@gmail.com

Choose an option:
1. Add a contact
2. Delete a contact
3. Display all contacts
4. Exit
Enter your choice: 4
Goodbye!
arkosh@greyskull:
~/repos/601_201/module11$ exit
exit

Script done on 2025-04-02 02:06:30-04:00 [COMMAND_EXIT_CODE="0"]
```