ASSIGNMENT:-5

QUES:04

Project Title: Contact List Application

Problem Statement: Design and implement a contact list application that can store, manage, and retrieve contact information. The application should use Java's Collections Framework to efficiently manage a list of contacts in memory. Additionally, it must utilize File Handling to persist the contact data to a file, allowing the user to save their contacts and load them back into the application.

Project Objectives:

- Understand and implement Java's File Handling capabilities.
- Use File class methods to manage files and directories.
- Implement reading from and writing to files using character streams.
- Apply the Java Collections Framework to store and manage data.
- Demonstrate the use of HashMap for efficient data storage and retrieval.

Learning Outcomes:

- Gain practical experience with Java I/O streams and file handling.
- Implement real-world applications using the Java Collections Framework.
- Develop robust applications that can persist data between sessions.

Project Instructions:

1. Contact Class:
   - Create a Contact class with attributes: name (String), phoneNumber (String), and email (String).
2. ContactManager Class:
   - Use a HashMap<String, Contact> to store contacts, where the key is the contact's name.
   - Implement methods to add, remove, and search for contacts.
   - Implement methods to saveContactsToFile(String filename) and loadContactsFromFile(String filename) using FileWriter and FileReader (or BufferedWriter and BufferedReader).
3. Main Class (ContactListApp):
   - Implement a menu-driven interface that allows the user to:
     - ♣ Add a new contact.
     - ♣ Remove an existing contact.
     - ♣ Search for a contact by name.
     - ♣ Display all contacts.
     - ♣ Save contacts to a file.
     - ♣ Load contacts from a file.
     - ♣ Exit the application.

Sample Interaction:

```
Welcome to the Contact List Application!
1. Add Contact
2. Search Contact
3. Display All Contacts
CO65. 4. Save to File
5. Load from File
6. Exit
Enter your choice: 1
Enter contact name: Alice
Enter phone number: 9876543210
Enter email: alice@example.com
Contact added successfully.
```

ANS:04

ASSIGNMENT:-5

ContactListApp.java

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

public class ContactListApp {

    // ---------------- Contact Class ----------------
    static class Contact {
        private String name;
        private String phoneNumber;
        private String email;

        public Contact(String name, String phoneNumber, String email) {
            this.name = name;
            this.phoneNumber = phoneNumber;
            this.email = email;
        }

        public String getName() {
            return name;
        }

        public String getPhoneNumber() {
            return phoneNumber;
        }

        public String getEmail() {
            return email;
        }

        @Override
        public String toString() {
            return "Name: " + name + ", Phone: " + phoneNumber + ", Email: " + email;
        }
    }

    // ---------------- Contact Manager Class ----------------
    static class ContactManager {

        private HashMap<String, Contact> contactMap = new HashMap<>();

        // Add Contact
        public void addContact(Contact contact) {
            contactMap.put(contact.getName(), contact);
            System.out.println("Contact added successfully!");
        }

        // Remove Contact
        public void removeContact(String name) {
            if (contactMap.remove(name) != null)
                System.out.println("Contact removed successfully!");
            else
                System.out.println("Contact not found!");
        }

        // Search Contact
        public Contact searchContact(String name) {
            return contactMap.get(name);
        }

```

ASSIGNMENT:-5

ContactListApp.java

```
60          // Display All Contacts
61          public void displayAllContacts() {
62              if (contactMap.isEmpty()) {
63                  System.out.println("No contacts available.");
64                  return;
65              }
66              for (Contact c : contactMap.values()) {
67                  System.out.println(c);
68              }
69          }
70
71          // Save Contacts
72          public void saveContactsToFile(String filename) {
73              try (BufferedWriter writer = new BufferedWriter(new FileWriter
                    (filename))) {
74
75                  for (Contact c : contactMap.values()) {
76                      writer.write(c.getName() + "," + c.getPhoneNumber() + "," +
                            c.getEmail());
77                      writer.newLine();
78                  }
79                  System.out.println("Contacts saved successfully!");
80
81              } catch (IOException e) {
82                  System.out.println("Error saving contacts: " + e.getMessage());
83              }
84          }
85
86          // Load Contacts
87          public void loadContactsFromFile(String filename) {
88              try (BufferedReader reader = new BufferedReader(new FileReader
                    (filename))) {
89
90                  contactMap.clear(); // clear old data
91                  String line;
92
93                  while ((line = reader.readLine()) != null) {
94                      String[] data = line.split(",");
95                      if (data.length == 3) {
96                          Contact c = new Contact(data[0], data[1], data[2]);
97                          contactMap.put(data[0], c);
98                      }
99                  }
100
101                 System.out.println("Contacts loaded successfully!");
102
103             } catch (IOException e) {
104                 System.out.println("Error loading contacts: " + e.getMessage());
105             }
106         }
107     }
```

ASSIGNMENT:-5

ContactListApp.java

Share Run

```
    // ---------- Main Method (Menu) ----------
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ContactManager manager = new ContactManager();

        while (true) {
            System.out.println("\n----- Contact List Application -----");
            System.out.println("1. Add Contact");
            System.out.println("2. Remove Contact");
            System.out.println("3. Search Contact");
            System.out.println("4. Display All Contacts");
            System.out.println("5. Save Contacts to File");
            System.out.println("6. Load Contacts from File");
            System.out.println("7. Exit");
            System.out.print("Enter your choice: ");

            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter name: ");
                    String name = sc.nextLine();
                    System.out.print("Enter phone number: ");
                    String phone = sc.nextLine();
                    System.out.print("Enter email: ");
                    String email = sc.nextLine();
                    Contact c = new Contact(name, phone, email);
                    manager.addContact(c);
                    break;

                case 2:
                    System.out.print("Enter name to remove: ");
                    String removeName = sc.nextLine();
                    manager.removeContact(removeName);
                    break;

                case 3:
                    System.out.print("Enter name to search: ");
                    String searchName = sc.nextLine();
                    Contact found = manager.searchContact(searchName);
                    if (found != null)
                        System.out.println("Contact: " + found);
                    else
                        System.out.println("Contact not found!");
                    break;

                case 4:
                    manager.displayAllContacts();
                    break;
```

ASSIGNMENT:-5

```
                case 4:
                    manager.displayAllContacts();
                    break;

                case 5:
                    System.out.print("Enter filename to save (e.g., contacts.txt
                        ): ");
                    String saveFile = sc.nextLine();
                    manager.saveContactsToFile(saveFile);
                    break;

                case 6:
                    System.out.print("Enter filename to load: ");
                    String loadFile = sc.nextLine();
                    manager.loadContactsFromFile(loadFile);
                    break;

                case 7:
                    System.out.println("Exiting...");
                    System.exit(0);

                default:
                    System.out.println("Invalid choice!");
            }
        }
    }
}
```

OUTPUT

ASSIGNMENT:-5

```
----- Contact List Application -----
1. Add Contact
2. Remove Contact
3. Search Contact
4. Display All Contacts
5. Save Contacts to File
6. Load Contacts from File
7. Exit
Enter your choice: 1
Enter name: kirpa
Enter phone number: 9958xxxxx
Enter email: kirpa@example.com
Contact added successfully!

----- Contact List Application -----
1. Add Contact
2. Remove Contact
3. Search Contact
4. Display All Contacts
5. Save Contacts to File
6. Load Contacts from File
7. Exit
Enter your choice: 3
Enter name to search: kirpa
Contact: Name: kirpa, Phone: 9958xxxxx, Email: kirpa@example.com

----- Contact List Application -----
1. Add Contact
2. Remove Contact
3. Search Contact
4. Display All Contacts
5. Save Contacts to File
6. Load Contacts from File
7. Exit
Enter your choice: 4
Name: kirpa, Phone: 9958xxxxx, Email: kirpa@example.com

----- Contact List Application -----
1. Add Contact
2. Remove Contact
3. Search Contact
4. Display All Contacts
5. Save Contacts to File
6. Load Contacts from File
7. Exit
Enter your choice: 5
Enter filename to save (e.g., contacts.txt): kirpa.txt
Error saving contacts: kirpa.txt (Permission denied)

----- Contact List Application -----
1. Add Contact
2. Remove Contact
3. Search Contact
4. Display All Contacts
5. Save Contacts to File
6. Load Contacts from File
7. Exit
Enter your choice: 2
Enter name to remove: kirpa
Contact removed successfully!
```