

10/25/2024

# ASSIGNMENT:2

## OOP

**NAME:** Ayesha Tayyaba

**SAP ID:** 56546

**Department:** CS

Riphah International University

**TASK:**

```
import java.util.ArrayList;
import java.util.Scanner;
class Contact {
    private String fName, lName;
    private String[] phoneNum = new String[3];
    private String affiliation;
    private String occupation;
    private String note = "";
    private String dob;
    private boolean blocked = false;
    public Contact(String fName, String lName, String[] phoneNum, String affiliation, String occupation,
String note, String dob) {
        this.fName = fName;
        this.lName = lName;
        this.phoneNum = phoneNum;
        this.affiliation = affiliation;
        this.occupation = occupation;
        this.note = note;
        this.dob = dob;
    }
    public void setFName(String fName)
    {
        this.fName = fName;
    }
    public void setlName(String lName)
    {
        this.lName = lName;
    }
    public void setPhoneNum(String[] phoneNum)
```

```
{
    this.phoneNum = phoneNum;
}

public void setAffiliation(String affiliation)
{
    this.affiliation = affiliation;
}

public void setOccupation(String occupation)
{
    this.occupation = occupation;
}

public void setNote(String note)
{
    this.note = note;
}

public void setDob(String dob)
{
    this.dob = dob;
}

public void setBlocked(boolean blocked)
{
    this.blocked = blocked;
}

public String getfName()
{
    return fName;
}

public String getlName()
{
    return lName;
}
```

```

    }

    public String[] getPhoneNum()
    {
        return phoneNum;
    }

    public String getAffiliation()
    {
        return affiliation;
    }

    public String getOccupation()
    {
        return occupation;
    }

    public String getNote()
    {
        return note;
    }

    public String getDob()
    {
        return dob;
    }

    public boolean isBlocked()
    {
        return blocked;
    }

    public void replaceNumber(String oldNum, String newNum) {
        for (int i = 0; i < phoneNum.length; i++) {
            if (phoneNum[i] != null && phoneNum[i].equals(oldNum)) {
                phoneNum[i] = newNum;
            }
        }
    }

```

```

        break;
    }
}
}
}

class Directory {
    private final ArrayList<Contact> directory;

    public Directory() {
        this.directory = new ArrayList<>();
    }

    public void addContact(Contact c) {
        directory.add(c);
    }

    public void addContact(String fName, String lName, String[] phoneNum, String affiliation, String
    occupation, String note, String dob) {
        Contact newContact = new Contact(fName, lName, phoneNum, affiliation, occupation, note, dob);
        addContact(newContact);
    }

    public Contact search(String fName) {
        for (Contact contact : directory) {
            if (contact.getfName().equalsIgnoreCase(fName)) {
                return contact;
            }
        }
        return null;
    }

    public boolean deleteContact(String fName) {
        for (Contact contact : directory) {
            if (contact.getfName().equalsIgnoreCase(fName)) {

```

```

        directory.remove(contact);

        return true;
    }
}

return false;
}

public void blockContact(String fName) {
    Contact contact = search(fName);
    if (contact != null) {
        contact.setBlocked(true);
    }
}

public String displayAllContacts() {
    StringBuilder dirInfo = new StringBuilder("Phone Directory:\n");
    for (Contact contact : directory) {
        dirInfo.append("Name: ").append(contact.getfName()).append("
").append(contact.getlName()).append("\n");
        dirInfo.append("Phone Numbers: ");
        for (String num : contact.getPhoneNum()) {
            if (num != null) {
                dirInfo.append(num).append(" ");
            }
        }
        dirInfo.append("\nAffiliation: ").append(contact.getAffiliation())
            .append("\nOccupation: ").append(contact.getOccupation())
            .append("\nNote: ").append(contact.getNote())
            .append("\nDOB: ").append(contact.getDob())
            .append("\nBlocked: ").append(contact.isBlocked()).append("\n\n");
    }
    return dirInfo.toString();
}

```

```

    }
}

public class TestApplication {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Directory directory = new Directory();
        while (true) {
            System.out.println("Menu:");
            System.out.println("1. Add Contact");
            System.out.println("2. Search Contact");
            System.out.println("3. Delete a Contact");
            System.out.println("4. Replace a Number");
            System.out.println("5. Display All");
            System.out.println("6. Block a Number");
            System.out.println("7. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1 -> {
                    System.out.print("Enter First Name: ");
                    String fName = scanner.nextLine();
                    System.out.print("Enter Last Name: ");
                    String lName = scanner.nextLine();
                    System.out.print("Enter Phone Numbers (comma-separated): ");
                    String[] phoneNum = scanner.nextLine().split(",");
                    System.out.print("Enter Affiliation: ");
                    String affiliation = scanner.nextLine();
                    System.out.print("Enter Occupation: ");

```

```

        String occupation = scanner.nextLine();
        System.out.print("Enter Note: ");
        String note = scanner.nextLine();
        System.out.print("Enter DOB (yyyy-mm-dd): ");
        String dob = scanner.nextLine();
        directory.addContact(fName, lName, phoneNum, affiliation, occupation, note, dob);
    }
case 2 -> {
    System.out.print("Enter First Name to Search: ");
    Contact foundContact = directory.search(scanner.nextLine());
    if (foundContact != null) {
        System.out.println("Name: " + foundContact.getfName() + " " +
foundContact.getlName());
        System.out.print("Phone Numbers: ");
        for (String num : foundContact.getPhoneNum()) {
            if (num != null) {
                System.out.print(num + " ");
            }
        }
        System.out.println("\nAffiliation: " + foundContact.getAffiliation());
        System.out.println("Occupation: " + foundContact.getOccupation());
        System.out.println("Note: " + foundContact.getNote());
        System.out.println("DOB: " + foundContact.getDob());
        System.out.println("Blocked: " + foundContact.isBlocked());
    } else {
        System.out.println("Contact not found.");
    }
}
case 3 -> {
    System.out.print("Enter First Name to Delete: ");

```



```

        if (directory.deleteContact(scanner.nextLine())) {
            System.out.println("Contact deleted.");
        } else {
            System.out.println("Contact not found.");
        }
    }
}

case 4 -> {
    System.out.print("Enter First Name to Replace Number: ");
    String nameToReplace = scanner.nextLine();
    Contact contactToReplace = directory.search(nameToReplace);
    if (contactToReplace != null) {
        System.out.print("Enter Old Number: ");
        String oldNumber = scanner.nextLine();
        System.out.print("Enter New Number: ");
        String newNumber = scanner.nextLine();
        contactToReplace.replaceNumber(oldNumber, newNumber);
        System.out.println("Number replaced.");
    } else {
        System.out.println("Contact not found.");
    }
}

case 5 ->
    System.out.println(directory.displayAllContacts());

case 6 ->
    System.out.print("Enter First Name to Block: ");
    directory.blockContact(scanner.nextLine());
    System.out.println("Contact blocked.");
}

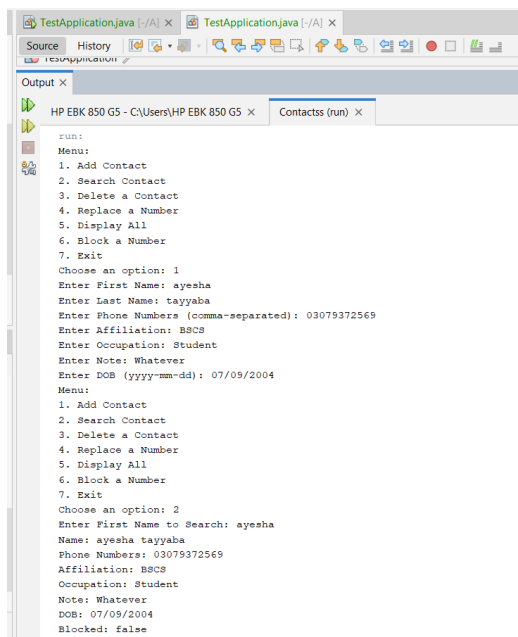
```

```

        case 7 -> {
            System.out.println("Exiting the application.");
            scanner.close();
            return;
        }
        default -> System.out.println("Invalid choice. Please try again.");
    }
}
}
}
}

```

## OUTPUT:



```

Menu:
1. Add Contact
2. Search Contact
3. Delete a Contact
4. Replace a Number
5. Display All
6. Block a Number
7. Exit
Choose an option: 7
Exiting the application.
BUILD SUCCESSFUL (total time: 2 minutes 0 seconds)
|

```