Name of the Course : Java 9 Core principles

Level : Difficult

Tool Stack : List Interface, Arraylist, Encapsulation and access specifiers

Problem Statement :

Airone mobile services needs to store their customer details in the company portal.The details are customer's first and last name, phone number, and email address. Help the company develop an application to maintain the details of their customer systematically.

Description : You are provided with a class Contact with the following attributes as private.

String firstName

String lastName

long phoneNumber

String emailId

A 4 argument constructor and appropriate setters and getters to store and retrieve the details are

also provided.

Create a class PhoneBook with a private attribute

List<Contact> phoneBook = new ArrayList<Contact>();

Write the getters and setters.

Write the following methods in the PhoneBook class.

public void addContact(Contact contactObj) – This method should add the contact object to the phoneBook

list.

public List<Contact> viewAllContacts() – This method should return the list of all contacts available.

public Contact viewContactGivenPhone(long phoneNumber) - This method should return the contact details

which has the phoneNumber given as parameter.

public boolean removeContact(long phoneNumber) - This method should remove the contact details which has

the phoneNumber given as parameter. If removed return true. Else if the phone number is

not available return false.

Write a class Main with the main method. Create the menu as shown in the Sample Input and Output and

invoke the corresponding methods as per the choice provided.

**Code:**

**package** main.java.yaksha;

**public** **class** Contact {

**private** String firstName;

**private** String lastName;

**private** **long** phoneNumber;

**private** String emailId;

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** **long** getPhoneNumber() {

**return** phoneNumber;

}

**public** **void** setPhoneNumber(**long** phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

**public** String getEmailId() {

**return** emailId;

}

**public** **void** setEmailId(String emailId) {

**this**.emailId = emailId;

}

**public** Contact(String firstName,String lastName,**long** phoneNumber,String emailId) {

**this**.firstName=firstName;

**this**.lastName=lastName;

**this**.phoneNumber=phoneNumber;

**this**.emailId=emailId;

}

}

package main.java.yaksha;

import java.util.ArrayList;

import java.util.List;

public class PhoneBook {

private List<Contact> phoneBook = new ArrayList<Contact>();

public List<Contact> getPhoneBook() {

return phoneBook;

}

public void setPhoneBook(List<Contact> phoneBook) {

this.phoneBook = phoneBook;

}

public boolean addContact(Contact contactObj) {

List<Contact> c=getPhoneBook();

c.add(contactObj);

setPhoneBook(c);

return true;

}

public List<Contact> viewAllContacts(){

return phoneBook;

}

public Contact viewContactGivenPhone(long phoneNumber) {

List<Contact> c2=getPhoneBook();

for(Contact list1:c2) {

if(list1.getPhoneNumber()==phoneNumber) {

return list1;

}

}

return (Contact) phoneBook;

}

public boolean removeContact(long phoneNumber) {

List<Contact> c3=getPhoneBook();

boolean status=false;

for(Contact list1:c3) {

if(list1.getPhoneNumber()==phoneNumber) {

status=c3.remove(list1);

return status;

}

}

return status;

}

}

package main.java.yaksha;

import java.util.Iterator;

import java.util.List;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

PhoneBook ph=new PhoneBook();

Scanner sc=new Scanner(System.in);

int ch;

do{

System.out.println("====Menu====");

System.out.println("1.Add Contact");

System.out.println("2.Display all contacts");

System.out.println("3.Search contact by phone");

System.out.println("4.Remove contact");

System.out.println("5.Exit");

System.out.println("Enter your choice: ");

ch=sc.nextInt();

switch(ch) {

case 1:

System.out.println("Add Contact: ");

System.out.println("Enter the First Name: ");

String firstName=sc.next();

System.out.println("Enter the Last Name: ");

String lastName=sc.next();

System.out.println("Enter the Phone No.:");

long phoneNumber=sc.nextLong();

System.out.println("Enter the Email:");

String emailId=sc.next();

Contact c=new Contact(firstName, lastName, phoneNumber, emailId);

ph.addContact(c);

break;

case 2:

List<Contact> phoneBook =ph.viewAllContacts();

for(Contact con:phoneBook) {

System.out.println("First Name:"+con.getFirstName());

System.out.println("Last Name:"+con.getLastName());

System.out.println("Phone No.:"+con.getPhoneNumber());

System.out.println("Email: "+con.getEmailId());

}

break;

case 3:

System.out.println("Enter the Phone number to search contact:");

long searchNumber=sc.nextLong();

Contact c2=ph.viewContactGivenPhone(searchNumber);

System.out.println(c2.getFirstName());

System.out.println(c2.getLastName());

System.out.println(c2.getPhoneNumber());

System.out.println(c2.getEmailId());

break;

case 4:

boolean success=false;

System.out.println("Enter the Phone number to remove :");

long removeNumber=sc.nextLong();

System.out.println("Do you want to remove the contact (Y/N):");

String check=sc.next();

if(check.equals("Y"))

{

success=ph.removeContact(removeNumber);

}

if(success) {

System.out.println("The contact is successfully deleted.");

}else {

System.out.println("The phone number is not available" );

}

break;

case 5:

System.exit(0);

break;

}

}while(ch<5);

}

Junit Testing

**package** test.java.yaksha;

**import** main.java.yaksha.\*;

**import** **static** org.junit.Assert.*assertEquals*;

**import** **static** org.junit.Assert.*assertTrue*;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.junit.jupiter.api.Test;

**class** MainTest {

@Test

**void** testAddContact() **throws** IOException {

**long** phone=77886655;

Contact c=**new** Contact("Jhon", "Mathew",phone, "jn@gmail.com");

PhoneBook pb=**new** PhoneBook();

assertTrue(pb.addContact(c));

}

@Test

**void** testViewAllContacts() **throws** IOException {

PhoneBook pb=**new** PhoneBook();

Contact con=new Contact("Jhon", "Mathew",784578456, "jn@gmail.com");

List<Contact> c=pb.getPhoneBook();

List<Contact> c1=pb.viewAllContacts();

assertEquals(true, c==c1);

}

@Test

**void** testViewContactGivenPhone() **throws** IOException {

PhoneBook pb=**new** PhoneBook();

Contact c1=**new** Contact("Jhon", "Mathew",77886655, "jn@gmail.com");

pb.addContact(c1);

Contact c2=pb.viewContactGivenPhone(77886655);

assertSame( c2 , c1 );

}

@Test

**void** testRemoveContact() **throws** IOException {

**long** phone=77886655;

Contact c1=**new** Contact("Jhon", "Mathew",phone, "jn@gmail.com");

PhoneBook pb=**new** PhoneBook();

pb.addContact(c1);

assertTrue(pb.removeContact(77886655));

}

}

Test Data1

====Menu====

1.Add Contact

2.Display all contacts

3.Search contact by phone

4.Remove contact

5.Exit

Enter your choice:

1

Add Contact:

Enter the First Name:

Jhon

Enter the Last Name:

Mathew

Enter the Phone No.:

77886655

Enter the Email:

jn@gmail.com

====Menu====

1.Add Contact

2.Display all contacts

3.Search contact by phone

4.Remove contact

5.Exit

Enter your choice:

1

Add Contact:

Enter the First Name:

Rama

Enter the Last Name:

Boke

Enter the Phone No.:

7898986899

Enter the Email:

rama@gmail.com

====Menu====

1.Add Contact

2.Display all contacts

3.Search contact by phone

4.Remove contact

5.Exit

Enter your choice:

2

First Name:Jhon

Last Name:Mathew

Phone No.:77886655

Email: jn@gmail.com

First Name:Rama

Last Name:Boke

Phone No.:7898986899

Email: rama@gmail.com

====Menu====

1.Add Contact

2.Display all contacts

3.Search contact by phone

4.Remove contact

5.Exit

Enter your choice:

3

Enter the Phone number to search contact:

7898986899

Rama

Boke

7898986899

rama@gmail.com

====Menu====

1.Add Contact

2.Display all contacts

3.Search contact by phone

4.Remove contact

5.Exit

Enter your choice:

4

Enter the Phone number to remove :

7898986899

Do you want to remove the contact (Y/N):

Y

The contact is successfully deleted.

====Menu====

1.Add Contact

2.Display all contacts

3.Search contact by phone

4.Remove contact

5.Exit

Enter your choice:

5

Learning outcome: Participant could able to learn how to use List Interface, add, remove, search operation on arraylist object , Encapsulation and access specifiers

.