//TESTAPPLICATION CLASS

package testapplication;

import java.util.GregorianCalendar;

import java.util.Scanner;

public class TestApplication {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Directory em = new Directory();

while (true) {

System.out.println("===== M E N U =====");

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

System.out.println("2. Search Contact");

System.out.println("3. Delete Contact");

System.out.println("4. Replace Contact");

System.out.println("5. Display All");

System.out.println("6. Block Contact");

System.out.println("0. Exit");

System.out.println(" ");

System.out.println("----------------------");

System.out.print(" Enter Your Choice: ");

int choice = sc.nextInt();

sc.nextLine();

System.out.println("----------------------");

switch (choice) {

case 1:

System.out.print("Enter Full Name: ");

String fName = sc.next();

String lName = sc.next();

sc.nextLine();

System.out.print("Enter Phone Numbers (comma separated): ");

String[] phoneNum = sc.nextLine().split(",");

System.out.print("Enter Affiliation: ");

String affiliation = sc.nextLine();

System.out.print("Enter Occupation: ");

String occupation = sc.nextLine();

System.out.print("Enter Note: ");

String note = sc.nextLine();

System.out.print("Enter Birth Year: ");

int year = sc.nextInt();

System.out.print("Enter Birth Month (1-12): ");

int month = sc.nextInt() - 1;

System.out.print("Enter Birth Day: ");

int day = sc.nextInt();

sc.nextLine();

GregorianCalendar dob = new GregorianCalendar(year, month, day);

em.addContact(fName, lName, phoneNum, affiliation, occupation, note, dob);

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT ADDED SUCCESSFULLY >>>");

sc.nextLine();

break;

case 2:

System.out.print("Enter First Name to Search: ");

fName = sc.nextLine();

Contact contact = em.searchContact(fName);

if (contact != null){

sc.nextLine();

System.out.print("RESULT: ");

System.out.println(contact);

sc.nextLine();

}

else{

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT NOT FOUND >>>");

sc.nextLine();

}

break;

case 3:

System.out.print("Enter First Name to Delete: ");

fName = sc.nextLine();

contact = em.searchContact(fName);

if (contact != null) {

boolean deleted = em.deleteContact(fName);

System.out.println("RESULT: ");

System.out.println("<<< CONTACT DELETED SUCCESSFULLY >>>");

}

else{

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT NOT FOUND >>>");

sc.nextLine();

}

break;

case 4:

System.out.print("Enter First Name for Replacement: ");

fName = sc.nextLine();

contact = em.searchContact(fName);

if (contact != null) {

System.out.print("Enter Old Number: ");

String oldNum = sc.nextLine();

System.out.print("Enter New Number: ");

String newNum = sc.nextLine();

contact.replaceNumber(oldNum, newNum);

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT REPLACED SUCCESSFULLY >>>");

sc.nextLine();

} else {

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT NOT FOUND >>>");

sc.nextLine();

}

break;

case 5:

System.out.println("ALL CONTACTS:\n");

System.out.println(em);

break;

case 6:

System.out.print("Enter First Name to Block: ");

fName = sc.nextLine();

contact = em.searchContact(fName);

if (contact != null) {

contact.setBlocked(true);

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT BLOCKED SUCCESSFULLY >>>");

sc.nextLine();

} else {

sc.nextLine();

System.out.println("RESULT: ");

System.out.println("<<< CONTACT NOT FOUND >>>");

sc.nextLine();

}

break;

case 0:

System.out.println("EXITING...");

return;

default:

System.out.println("Invalid Choice. Please enter a number between 0 and 6.");

}

}

}

}

// CONTACT CLASS

package testapplication;

import java.util.Arrays;

import java.util.GregorianCalendar;

public class Contact {

private String fName, lName;

private String[] phoneNum;

private String affiliation;

private String occupation;

private String note = "";

private GregorianCalendar dob;

private boolean blocked;

// Default Constructor

public Contact() {

}

// Parameterized Constructor

public Contact(String fName, String lName, String[] phoneNum, String affiliation, String occupation, String note, GregorianCalendar dob, boolean blocked) {

this.fName = fName;

this.lName = lName;

this.phoneNum = phoneNum;

this.affiliation = affiliation;

this.occupation = occupation;

this.note = note;

this.dob = dob;

this.blocked = blocked;

}

// Overloaded Constructor

public Contact(String fName, String lName, String[] phoneNum) {

this.fName = fName;

this.lName = lName;

this.phoneNum = phoneNum;

}

// Getters and Setters

public String getFName() {

return fName;

}

public void setFName(String fName) {

this.fName = fName;

}

public String getLName() {

return lName;

}

public void setLName(String lName) {

this.lName = lName;

}

public String[] getPhoneNum() {

return phoneNum;

}

public void setPhoneNum(String[] phoneNum) {

this.phoneNum = phoneNum;

}

public String getAffiliation() {

return affiliation;

}

public void setAffiliation(String affiliation) {

this.affiliation = affiliation;

}

public String getOccupation() {

return occupation;

}

public void setOccupation(String occupation) {

this.occupation = occupation;

}

public String getNote() {

return note;

}

public void setNote(String note) {

this.note = note;

}

public GregorianCalendar getDob() {

return dob;

}

public void setDob(GregorianCalendar dob) {

this.dob = dob;

}

public boolean isBlocked() {

return blocked;

}

public void setBlocked(boolean blocked) {

this.blocked = blocked;

}

// Method to Replace Number

public boolean replaceNumber(String oldNum, String newNum) {

for (int i = 0; i < phoneNum.length; i++) {

if (phoneNum[i] != null && phoneNum[i].equals(oldNum)) {

phoneNum[i] = newNum;

return true;

}

}

return false;

}

// toString() Method

@Override

public String toString() {

return "Name: " + fName + " " + lName + "\nContact Number: " + Arrays.toString(phoneNum) + "\nAffiliation: " + affiliation + "\nOccupation: " + occupation + "\nNote: " + note + "\nDOB: " + dob.getTime() + "\nBlocked? " + blocked;

}

}

// DIRECTORY CLASS

package testapplication;

import java.util.ArrayList;

import java.util.GregorianCalendar;

public class Directory {

private ArrayList<Contact> directory;

private int num;

// Default Constructor

public Directory() {

this.directory = new ArrayList<>();

this.num = 0;

}

// Parameterized Constructor

public Directory(ArrayList<Contact> directory, int num) {

this.directory = directory;

this.num = num;

}

// Overload Constructor

public Directory(int num) {

this.directory = new ArrayList<>();

this.num = num;

}

// Getters and Setters

public ArrayList<Contact> getDirectory() {

return directory;

}

public void setDirectory(ArrayList<Contact> directory) {

this.directory = directory;

}

public int getNum() {

return num;

}

public void setNum (int num){

this.num = num;

}

// toString() Method

@Override

public String toString() {

String result = "";

for (Contact contact : directory) {

result += contact.toString() + "\n";

}

return result;

}

// Add Contact

public void addContact(Contact c) {

directory.add(c);

num++;

}

// Another Method To Add Contact

public void addContact(String fName, String lName, String[] phoneNum, String affiliation, String occupation, String note, GregorianCalendar dob) {

Contact contact = new Contact(fName, lName, phoneNum, affiliation, occupation, note, dob, false);

addContact(contact);

}

// Search a contact

public Contact searchContact(String fName) {

for (Contact contact : directory) {

if (contact.getFName().equalsIgnoreCase(fName)) {

return contact;

}

}

return null;

}

// Delete a contact

public boolean deleteContact(String fName) {

Contact contact = searchContact(fName);

if (contact != null) {

directory.remove(contact);

return true;

}

return false;

}

}