**DSA project: phonebook**

Modules:

* Contacts
* Search (Contacts)
* Display (Contacts)
* Update (Contacts)
* Sort (Contacts)

Functions:

1. **Insert Contact**: client (user) will insert new contacts and keep them stored in the Phonebook of their mobile devices.
2. **Search Contact**: client (user) will be able to search for contacts throughout the phonebook using linear Stack apllications
3. **Display Contact**: client (user) will be to display all contacts in their phonebook
4. **Deletion Contact**: client (user) will be able to delete any contact wished in their phonebook
5. **Update Contact**: client (user) can update by means of adding more information to the desired contact they wish or remove information desired
6. **Sort Contacts**: client (user) can arrange their contacts into their desired arrangement they wish by example of either sorting in alphabetical order or time user inserted their contacts etc.

Pseudocodes for the modules:

**Insert contact**

Start

Prompt for name and number

Get name and number

Number = digits

IF (digits > 10) then

Display “number does not exist”

Else IF (digits = 10)

Display “number is added”

Endif

End

**Search contact**

Start

Prompt for number

Get number

Top = temp

IF (top == null) then

Display “Phonebook is empty”

Else

Display “number”

Endif

End

**Display all contacts**

Display () {

top = temp

IF (top == null) Then

Display “Phonebook is empty”

Else

While (temp! = null)

Display temp -> number

temp = temp – next

Endwhile

Endif

}

**Delete contact**

Start

Prompt for number

Get number

IF (top == null) Then

Display “Phonebook is empty”

ELSE

number = Phonebook[top]

top --

Display “The deleted number is:” + number

Endif

End

**Update contact**

Start

Prompt for number and update

Get number and update

IF (top == null) Then

Display “Phonebook is empty”

Else

Display “contact is updated”

Endif

End

**Sort contact**

Start

For (i = 1; i < n; i++)

List of numbers = i

temp = array[i]

j = i - 1

While (j >= 0 AND array[j] > temp)

array [j + 1] = array[j]

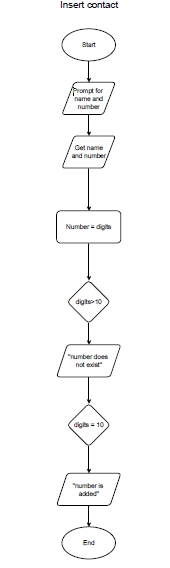
j = j – 1

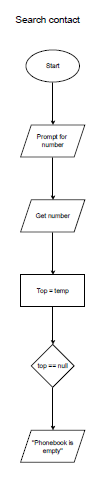
Endwhile

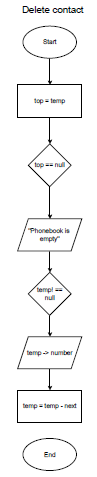
array [j + 1] = temp

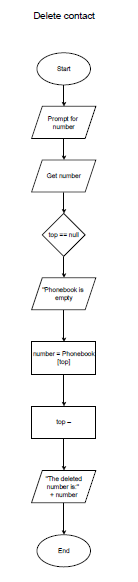
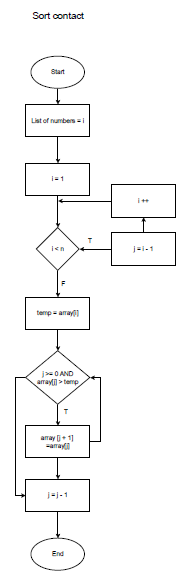
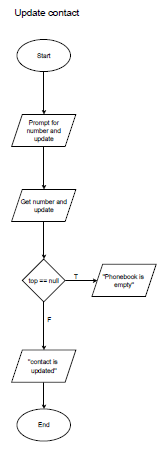
Endfor

End







import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Scanner;

class Contact {

String name;

String phoneNumber;

public Contact(String name, String phoneNumber) {

this.name = name;

this.phoneNumber = phoneNumber;

}

@Override

public String toString() {

return "Name: " + name + ", Phone Number: " + phoneNumber;

}

}

class Phonebook {

private ArrayList<Contact> contacts;

public Phonebook() {

contacts = new ArrayList<>();

}

public void insertContact(String name, String phoneNumber) {

contacts.add(new Contact(name, phoneNumber));

System.out.println("Contact added: " + name);

displayAllContacts();

}

public Contact searchContact(String name) {

for (Contact contact : contacts) {

if (contact.name.equalsIgnoreCase(name)) {

System.out.println("Found: " + contact);

return contact;

}

}

System.out.println("Contact not found.");

return null;

}

public void displayAllContacts() {

if (contacts.isEmpty()) {

System.out.println("Phonebook is empty.");

return;

}

System.out.println("\nAll Contacts:");

for (Contact contact : contacts) {

System.out.println(contact);

}

}

public void deleteContact(String name) {

Contact contactToRemove = null;

for (Contact contact : contacts) {

if (contact.name.equalsIgnoreCase(name)) {

contactToRemove = contact;

break;

}

}

if (contactToRemove != null) {

contacts.remove(contactToRemove);

System.out.println("Contact deleted: " + name);

} else {

System.out.println("Contact not found.");

}

displayAllContacts();

}

public void updateContact(String name, String newPhoneNumber) {

for (Contact contact : contacts) {

if (contact.name.equalsIgnoreCase(name)) {

contact.phoneNumber = newPhoneNumber;

System.out.println("Contact updated: " + name);

displayAllContacts();

return;

}

}

System.out.println("Contact not found.");

}

public void sortContacts() {

Collections.sort(contacts, new Comparator<Contact>() {

@Override

public int compare(Contact c1, Contact c2) {

return c1.name.compareToIgnoreCase(c2.name);

}

});

System.out.println("Contacts sorted.");

displayAllContacts();

}

}

public class PhonebookApp {

public static void main(String[] args) {

Phonebook phonebook = new Phonebook();

Scanner scanner = new Scanner(System.in);

String command;

do {

System.out.println("\nPhonebook Menu:");

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

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

System.out.println("3. Display All Contacts");

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

System.out.println("5. Update Contact");

System.out.println("6. Sort Contacts");

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

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

command = scanner.nextLine();

switch (command) {

case "1":

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

String name = scanner.nextLine();

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

String phoneNumber = scanner.nextLine();

phonebook.insertContact(name, phoneNumber);

break;

case "2":

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

}

}

public class PhonebookApp {

public static void main(String[] args) {

Phonebook phonebook = new Phonebook();

Scanner scanner = new Scanner(System.in);

String command;

do {

System.out.println("\nPhonebook Menu:");

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

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

System.out.println("3. Display All Contacts");

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

System.out.println("5. Update Contact");

System.out.println("6. Sort Contacts");

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

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

command = scanner.nextLine();

switch (command) {

case "1":

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

String name = scanner.nextLine();

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

String phoneNumber = scanner.nextLine();

phonebook.insertContact(name, phoneNumber);

break;

case "2":

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

}