**DSA project: phonebook**

**Contributors:**

* **Frieda IM Ngiikumbu : 223012246**
* **Ausiku Asteria N : 224080563**
* **Kwedhi Gabriel : 224014056**
* **Wayne Feris : 224016466**
* **Jayden Claassen : 224085832**
* **Immanuel Konjore : 224016660**

Modules:

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

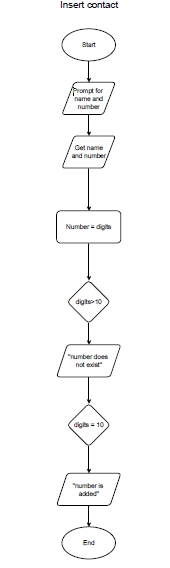
j = j – 1

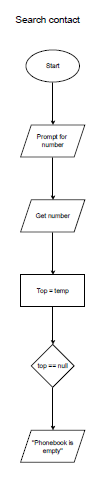
Endwhile

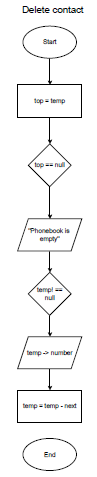
array [j + 1] = temp

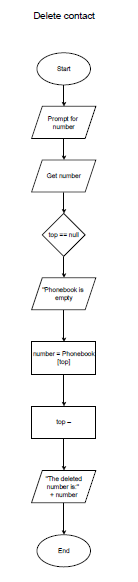
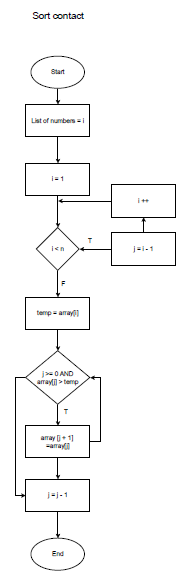
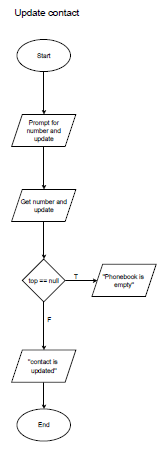
Endfor

End







DSA PROJECT CODE: source code

Import java.util.ArrayList;

Import java.util.Collections;

Import java.util.Comparator;

Import java.util.Scanner;

Class Contact {

String name;

String number;

Contact(String name, String number) {

This.name = name;

This.number = number;

}

@Override

Public String toString() {

Return “Name: “ + name + “, Number: “ + number;

}

}

Public class Phonebook {

Private ArrayList<Contact> contacts = new ArrayList<>();

Private Scanner scanner = new Scanner(System.in);

// Insert Contact

Public void insertContact() {

System.out.print(“Enter name: “);

String name = scanner.nextLine();

System.out.print(“Enter number: “);

String number = scanner.nextLine();

If (number.length() > 10) {

System.out.println(“Number does not exist.”);

} else if (number.length() == 10) {

Contacts.add(new Contact(name, number));

System.out.println(“Number is added.”);

}

}

// Search Contact

Public void searchContact() {

System.out.print(“Enter number to search: “);

String searchNumber = scanner.nextLine();

Long startTime = System.nanoTime();

For (Contact contact : contacts) {

If (contact.number.equals(searchNumber)) {

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

Long endTime = System.nanoTime();

System.out.println(“Search time: “ + (endTime – startTime) + “ ns”);

Return;

}

}

Long endTime = System.nanoTime();

System.out.println(“Contact not found. Search time: “ + (endTime – startTime) + “ ns”);

}

// Display All Contacts

Public void displayContacts() {

If (contacts.isEmpty()) {

System.out.println(“Phonebook is empty.”);

Return;

}

For (Contact contact : contacts) {

System.out.println(contact);

}

}

// Delete Contact

Public void deleteContact() {

System.out.print(“Enter number to delete: “);

String deleteNumber = scanner.nextLine();

For (int i = 0; i < contacts.size(); i++) {

If (contacts.get(i).number.equals(deleteNumber)) {

System.out.println(“The deleted number is: “ + contacts.get(i).number);

Contacts.remove(i);

Return;

}

}

System.out.println(“Phonebook is empty or contact not found.”);

}

// Update Contact

Public void updateContact() {

System.out.print(“Enter number to update: “);

String updateNumber = scanner.nextLine();

For (Contact contact : contacts) {

If (contact.number.equals(updateNumber)) {

System.out.print(“Enter new name: “);

String newName = scanner.nextLine();

Contact.name = newName; // Update name

System.out.println(“Contact is updated.”);

Return;

}

}

System.out.println(“Phonebook is empty or contact not found.”);

}

// Sort Contacts

Public void sortContacts() {

Collections.sort(contacts, Comparator.comparing(c -> c.number));

System.out.println(“Contacts sorted successfully.”);

}

Public static void main(String[] args) {

Phonebook phonebook = new Phonebook();

While (true) {

System.out.println(“\n1. 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(“7. Exit”);

System.out.print(“Choose an option: “);

Int choice = phonebook.scanner.nextInt();

Phonebook.scanner.nextLine(); // Consume newline

Switch (choice) {

Case 1:

Phonebook.insertContact();

Break;

Case 2:

Phonebook.searchContact();

Break;

Case 3:

Phonebook.displayContacts();

Break;

Case 4:

Phonebook.deleteContact();

Break;

Case 5:

Phonebook.updateContact();

Break;

Case 6:

Phonebook.sortContacts();

Break;

Case 7:

System.exit(0);

Break;

Default:

System.out.println(“Invalid option. Please try again.”);

}

}

}

}