JAVA PROJECT  
Contact Book

short line

**G. Rakshitha ( AP19110010280 )**

**D. Uday Kiran ( AP19110010306 )**

**D. Sai Praween ( AP19110010215 )**

# **ALGORITHM:**

* Step 1: START
* Step 2: First we should create a class AddressBook( model object ) and declare the variables.
* Step 3:And declare variables phoneNumbers as a data type list String and email as a data type String and make them private ( which is viewed or used only in that particular class).
* Step 4: Create bers with data type List String and return PhoneNumbers.a method named getPhoneNum
* Step 5: Create a parameterised method setPhoneNumbers with data type void and use “ this ” keyword (which refers to the current object in a method or constructor) for phoneNumbers.
* Step 6: Create a method named getEmail with data type String and return email.
* Step 7: Create a parameterised method setEmail with data type void and use this keyword (which refers to the current object in a method or constructor) for email.
* Step 8: Create a method named toString with data type String and return AddressBook containing phone number and email.
* Step 9: Now create a new class named AddressBookDemo import packages and header files which are needed by the code.
* Step 10: In the class AddressBookDemo declare the phone\_book\_demo\_new (name of the file in which the phone book data is stored) with data type string and make it private which is viewed or used only in that particular class.
* Step 11: In the main function declare the variables name, number and email with data type String.
* Step 12: And create a Treemap data structure(it stores the values as pairs) named phonebook for storing the values name, email and number. We store the name and AddressBook(contain email,number) to the phone book .
* Step 13: Now create a dataFile variable of type File to represent the data file that is stored in the user's home directory.
* Step 14: If the data file does not exist it prints no data file found and creates the new file. If the data file already exists, then the data in the file is read and is used to initialize the phone directory.
* Step 15: The format of the file follows the way each line of the file represents one directory entry, with the name , email and the number for that entry separated by the character '%'.
* Step 16: If a file exists but does not have this format, then the program terminates (this is done to avoid overwriting a file that is being used for another purpose).
* Step 17: Now we use the switch case to read commands from the user

Case 1. Search the phone number using the name .

Case 2. Search the phone number using the email .

Case 3. Add or edit phone number to the phone directory.

Case 4. Remove an entry from your phone directory.

Case 5. Display the entire phone directory.

* Step 18: And carry them out, until the user gives the Exit from program command
* Step 19: Now write the current contents of the phone directory, but only if some changes have been made to the directory.
* Step 20: END

# **CODE:**

# import java.util.List;

# public class AddressBook {

# private List<String> phoneNumbers;

# private String email;

# public List<String> getPhoneNumbers() {

# return phoneNumbers;

# }

# public void setPhoneNumbers(List<String> phoneNumbers) {

# this.phoneNumbers = phoneNumbers;

# }

# public String getEmail() {

# return email;

# }

# public void setEmail(String email) {

# this.email = email;

# }

# @Override

# public String toString() {

# return "AddressBook [phoneNumbers=" + phoneNumbers + ", email=" + email + "]";

# }

# 

# }

# 

# 

# 

# 

# import java.io.File;

# import java.io.FileWriter;

# import java.io.IOException;

# import java.io.PrintWriter;

# import java.util.ArrayList;

# import java.util.List;

# import java.util.Map;

# import java.util.Scanner;

# import java.util.TreeMap;

# 

# /\*\*

# \* This program lets the user keep a persistent "phone book" that

# \* contains names,email and phone numbers. The data for the phone book

# \* is stored in a file in the user's home directory.

# \*/

# 

# public class AddressBookDemo {

# 

# /\*\*

# \* The name of the file in which the phone book data is kept. The

# \* file is stored in the user's home directory.

# \*/

# private static String DATA\_FILE\_NAME = ".phone\_book\_demo\_new";

# public static void main(String[] args) {

# String name, number,email;

# TreeMap<String, AddressBook> phoneBook; // Phone directory data structure // Enties are name/number pairs.

# phoneBook = new TreeMap<String,AddressBook>();

# File userHomeDirectory = new File( System.getProperty("user.home") );

# File dataFile = new File( userHomeDirectory, DATA\_FILE\_NAME );

# 

# 

# /\* If the data file already exists, then the data in the file is

# \* read and is used to initialize the phone directory.

# \*/

# 

# if ( ! dataFile.exists() ) {

# System.out.println("No phone book data file found.");

# System.out.println("A new one will be created.");

# System.out.println("File name: " + dataFile.getAbsolutePath());

# }

# else {

# System.out.println("Reading phone book data...");

# try {

# Scanner scanner = new Scanner( dataFile );

# while (scanner.hasNextLine()) {

# String phoneEntry = scanner.nextLine();

# int separatorPosition = phoneEntry.indexOf('%');

# if (separatorPosition == -1)

# throw new IOException("File is not a phonebook data file.");

# name = phoneEntry.substring(0, separatorPosition);

# number = phoneEntry.substring(separatorPosition+1);

# email = phoneEntry.substring(0,separatorPosition);

# AddressBook addressBook = new AddressBook();

# List<String> phoneNumberList = new ArrayList<>();

# phoneNumberList.add(number);

# addressBook.setPhoneNumbers(phoneNumberList);

# addressBook.setEmail(email);

# 

# phoneBook.put(name,addressBook);

# }

# }

# catch (IOException e) {

# System.out.println("Error in phone book data file.");

# System.out.println("File name: " + dataFile.getAbsolutePath());

# System.out.println("This program cannot continue.");

# System.exit(1);

# }

# }

# 

# 

# /\* Read commands from the user and carry them out, until the

# \* user gives the "Exit from program" command.

# \*/

# 

# Scanner in = new Scanner( System.in );

# boolean changed = false; // Have any changes been made to the directory?

# 

# mainLoop: while (true) {

# System.out.println("\nSelect the action that you want to perform:");

# System.out.println(" 1. Look up a phone number using name .");

# System.out.println(" 2. Look up a phone number using email .");

# System.out.println(" 3. Add or change a phone number.");

# System.out.println(" 4. Remove an entry from your phone directory.");

# System.out.println(" 5. List the entire phone directory.");

# System.out.println(" 6. Exit from the program.");

# System.out.println("Enter action number (1-6): ");

# int command;

# if ( in.hasNextInt() ) {

# command = in.nextInt();

# in.nextLine();

# }

# else {

# System.out.println("\nILLEGAL RESPONSE. YOU MUST ENTER A NUMBER.");

# in.nextLine();

# continue;

# }

# switch(command) {

# case 1:

# System.out.print("\nEnter the name whose number you want to look up: ");

# name = in.nextLine().trim().toLowerCase();

# AddressBook addressBook = phoneBook.get(name);

# 

# if (addressBook.getPhoneNumbers() == null)

# System.out.println("\nSORRY, NO NUMBER FOUND FOR " + name);

# else

# System.out.println("\nNUMBER FOR " + name + ": " + addressBook.getPhoneNumbers().get(0));

# 

# break;

# case 2:

# System.out.print("\nEnter the email whose number you want to look up: ");

# email = in.nextLine().trim().toLowerCase();

# boolean emailFound = false;

# String phoneNumber = null;

# for (Map.Entry<String, AddressBook> entry : phoneBook.entrySet()) {

# AddressBook addressBookFromMap = entry.getValue();

# 

# if(addressBookFromMap.getEmail() != null && addressBookFromMap.getEmail().equalsIgnoreCase(email)) {

# emailFound = true;

# phoneNumber = addressBookFromMap.getPhoneNumbers().get(0);

# break;

# }

# }

# 

# 

# if(emailFound)

# System.out.println("\nNUMBER FOR " + email + ": " + phoneNumber);

# else

# System.out.println("\nSorry No NUMBER FOUND FOR "+ email);

# break;

# case 3:

# System.out.print("\nEnter the name: ");

# name = in.nextLine().trim().toLowerCase();

# System.out.print("\nEnter the email: ");

# email = in.nextLine().trim().toLowerCase();

# if (name.length() == 0)

# System.out.println("\nNAME CANNOT BE BLANK.");

# else if (name.indexOf('%') >= 0)

# System.out.println("\nNAME CANNOT CONTAIN THE CHARACTER \"%\".");

# else if (email.length()==0)

# System.out.println("\nEMAIL CANNOT BE BLANK\n ");

# else if(!email.contains("@")) {

# System.out.println(" Email should contain @ character");

# }

# else {

# System.out.print("\nEnter phone number: ");

# number = in.nextLine().trim();

# if (number.length() == 0)

# System.out.println("\nPHONE NUMBER CANNOT BE BLANK.");

# else if(number.length()>10 || number.length()<10)

# System.out.println("\nPHONE NUMBER SHOULD CONTAIN 10 DIGITS");

# else {

# AddressBook newAddressBook = new AddressBook();

# List<String> phoneNumberList = new ArrayList<>();

# phoneNumberList.add(number);

# newAddressBook.setPhoneNumbers(phoneNumberList);

# newAddressBook.setEmail(email);

# 

# phoneBook.put(name,newAddressBook);

# System.out.println("\nContact Successfully added");

# changed = true;

# }

# }

# break;

# case 4:

# System.out.print("\nEnter the name whose entry you want to remove: ");

# name = in.nextLine().trim().toLowerCase();

# addressBook = phoneBook.get(name);

# if (addressBook == null)

# System.out.println("\nSORRY, THERE IS NO ENTRY FOR " + name);

# else {

# phoneBook.remove(name);

# changed = true;

# System.out.println("\nDIRECTORY ENTRY REMOVED FOR " + name);

# }

# break;

# case 5:

# System.out.println("\nLIST OF ENTRIES IN YOUR PHONE BOOK:\n");

# 

# for (Map.Entry<String, AddressBook> entry : phoneBook.entrySet()) {

# AddressBook addressBookFromMap = entry.getValue();

# System.out.println(" " + entry.getKey()+ ": " + addressBookFromMap.getPhoneNumbers().get(0)+ ": " + addressBookFromMap.getEmail());

# }

# break;

# case 6:

# System.out.println("\nExiting program.");

# break mainLoop;

# default:

# System.out.println("\nILLEGAL ACTION NUMBER.");

# }

# }

# 

# 

# /\* Before ending the program, write the current contents of the

# \* phone directory, but only if some changes have been made to

# \* the directory.

# \*/

# 

# if (changed) {

# System.out.println("Saving phone directory changes to file " +

# dataFile.getAbsolutePath() + " ...");

# PrintWriter out;

# try {

# out = new PrintWriter( new FileWriter(dataFile) );

# }

# catch (IOException e) {

# System.out.println("ERROR: Can't open data file for output.");

# return;

# }

# for (Map.Entry<String, AddressBook> entry : phoneBook.entrySet()) {

# AddressBook addressBookFromMap = entry.getValue();

# System.out.println(" " + entry.getKey()+ ": " + addressBookFromMap.getPhoneNumbers().get(0)+ ": " + addressBookFromMap.getEmail());

# out.println(entry.getKey() + "%" + addressBookFromMap.getPhoneNumbers().get(0) + "%" + addressBookFromMap.getEmail());

# }

# out.close();

# if (out.checkError())

# System.out.println("ERROR: Some error occurred while writing data file.");

# else

# System.out.println("Done.");

# }

# 

# }

# 

# }

# 

# 

# **OUTPUT:**

Reading phone book data….

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

3

Enter the name: priya

Enter the email: priya@gmail.com

Enter phone number: 90576334

PHONE NUMBER SHOULD CONTAIN 10 DIGITS

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

3

Enter the name: priya

Enter the email: priya@gmail.com

Enter phone number: 9057633457

Contact Successfully added

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

3

Enter the name: Srinivas

Enter the email: Srinivas@gmail.com

Enter phone number: 7305444878

Contact Successfully added

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

3

Enter the name: Riya

Enter the email: Riya@gmail.com

Enter phone number: 6345678223

Contact Successfully added

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

1

Enter the name whose number you want to look up: priya

NUMBER FOR priya: 9057633457

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

1

Enter the name whose number you want to look up: Srinivas

NUMBER FOR srinivas: 7305444878

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

1

Enter the name whose number you want to look up: Riya

NUMBER FOR riya: 6345678223

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

2

Enter the email whose number you want to look up: Riya@gmail.com

NUMBER FOR riya@gmail.com: 6345678223

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

5

LIST OF ENTRIES IN YOUR PHONE BOOK:

priya: 9057633457: priya@gmail.com

riya: 6345678223: riya@gmail.com

srinivas: 7305444878: srinivas@gmail.com

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

4

Enter the name whose entry you want to remove: Riya

DIRECTORY ENTRY REMOVED FOR riya

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

5

LIST OF ENTRIES IN YOUR PHONE BOOK:

priya: 9057633457: priya@gmail.com

srinivas: 7305444878: srinivas@gmail.com

Select the action that you want to perform:

1. Look up a phone number using name .

2. Look up a phone number using email .

3. Add or change a phone number.

4. Remove an entry from your phone directory.

5. List the entire phone directory.

6. Exit from the program.

Enter action number (1-6):

6

Exiting program.

Saving phone directory changes to file C:\Users\raksh\.phone\_book\_demo\_new ...

priya: 9057633457: priya@gmail.com

srinivas: 7305444878: srinivas@gmail.com

Done..

# **FLOWCHART:**

