1. A phone directory holds a list of names with a phone number for each name. It is possible to find the number associated with a given name, and to specify the phone number for a given name.[Hint:use a HashMap to store names as keys and phone numbers as values. The user can choose to find a number by providing a name or add a new name with a phone number. The program runs in a loop until the user chooses to exit.]

Solution:

import java.util.HashMap;

import java.util.Scanner;

public class PhoneDirectory {

public static void main(String[] args) {

HashMap<String, String> phoneDirectory = new HashMap<>();

Scanner scanner = new Scanner(System.in);

String choice;

do {

System.out.println("Phone Directory Menu:");

System.out.println("1. Find phone number by name");

System.out.println("2. Add new name with phone number");

System.out.println("3. Exit");

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

choice = scanner.nextLine();

switch (choice) {

case "1":

System.out.print("Enter name to find phone number: ");

String name = scanner.nextLine();

String phoneNumber = phoneDirectory.get(name);

if (phoneNumber != null) {

System.out.println("Phone number for " + name + " is " + phoneNumber);

} else {

System.out.println("Name not found in the directory.");

}

break;

case "2":

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

String newName = scanner.nextLine();

System.out.print("Enter phone number: ");

String newPhoneNumber = scanner.nextLine();

phoneDirectory.put(newName, newPhoneNumber);

System.out.println("New entry added successfully.");

break;

case "3":

System.out.println("Exiting the program.");

break;

default:

System.out.println("Invalid choice. Please try again.");

}

} while (!choice.equals("3"));

scanner.close();

}

}

Output:

Phone Directory Menu:

1. Find phone number by name

2. Add new name with phone number

3. Exit

Enter your choice: 2

Enter name: John

Enter phone number: 1234567890

New entry added successfully.

Phone Directory Menu:

1. Find phone number by name

2. Add new name with phone number

3. Exit

Enter your choice: 1

Enter name to find phone number: John

Phone number for John is 1234567890

Phone Directory Menu:

1. Find phone number by name

2. Add new name with phone number

3. Exit

Enter your choice: 3

Exiting the program.

1. Write a Java program to search a key in a Tree Map. [Hint:create a TreeMap and add some name-age pairs to it. Then, the user is prompted to enter a name to search for. If the name exists in the TreeMap, the corresponding age is displayed; otherwise, a message indicating that the name was not found is shown.]

Solution:

import java.util.Scanner;

import java.util.TreeMap;

public class TreeMapSearch {

public static void main(String[] args) {

TreeMap<String, Integer> treeMap = new TreeMap<>();

treeMap.put("Alice", 25);

treeMap.put("Bob", 30);

treeMap.put("Charlie", 35);

treeMap.put("David", 40);

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a name to search for: ");

String name = scanner.nextLine();

if (treeMap.containsKey(name)) {

int age = treeMap.get(name);

System.out.println("Age of " + name + " is " + age);

} else {

System.out.println("Name not found in the TreeMap.");

}

scanner.close();

}

}

Output:

Enter a name to search for: Bob

Age of Bob is 30

Enter a name to search for: Eve

Name not found in the TreeMap.