package PhoneBookIgorLanger;

import javax.print.attribute.standard.PresentationDirection;

import javax.swing.\*;

import java.util.\*;

import static java.util.Collections.\*;

public class ListContactsInFile {

/\*\*

\* This is a function that returns a list of all entries in the book.

\* In case there is no record in the book, a detailed message appears.

\* In addition, there is a function of sorting the name values in reverse order.

\*/

public static void listContactsInFile(Map<String, List<String>> contacts) {

System.out.println("Below are all the contacts that appear in the phone book in alphabetical order.");

System.out.println();

if (!contacts.isEmpty()) {

for (Map.Entry<String, List<String>> entry : contacts.entrySet()) {

if (contacts.equals(contacts.entrySet())) {

System.out.println("These are doubles values in phone book");

System.out.println(contacts.entrySet());

}

System.out.println("Name: " + entry.getKey());

for (String number : entry.getValue()) {

System.out.println("Number: " + number);

}

System.out.println();

}

} else {

System.out.println("No records found, the phone book is empty!");

}

try {

confirm:

while (true) {

System.out.println("Do you want arrangement the contacts in reverse order [Y/N] ?");

Scanner input = new Scanner(System.in);

String confirmation = input.nextLine().trim().toLowerCase();

switch (confirmation) {

case "y":

ArrayList<String> list = new ArrayList<String>();

list.add(contacts.entrySet().toString());

Collections.sort(list, reverseOrder());

System.out.println("Records of names after arrangement in reverse order");

System.out.println(list);

break confirm;

case "n":

System.out.println();

break confirm;

}

}

System.out.println("Type a command or 'exit' to quit. For a list of valid commands use 'help':");

} catch (Exception e) {

System.out.println("You typed an invalid parameter, returns to the main menu");

}

}

}