**Name: Maryam Abid**

**Regno: SP24-BSE-058**

**OOP ASIGNMENT NO 1(THEORY)**

**Messaging app**

**Classes:**

1. Sender.java
2. Message.java
3. Contacts.java
4. Demo.java

**Variables in the classes:**

**Sender.java**

public Message[] messages;

    public static int countercount = 0;

    public Contacts[][] proContacts;

    public static int index = 0;

Here I took the 1d array of Message class and 2d array of Contacts class.

**Message.java**

private String sender="Maryam";

    private String reciever;

    private String content;

    private String messageid;

    private  boolean send;

    private boolean seen;

    private String name;

    LocalDateTime time;

    private static int counter=0;

Here are all the attributes of Message class

LocalDateTime is the class of java I imported

import java.time.LocalDateTime;

**Contacts.java**

private String name;

    private String number;

**Demo.java**

Sender send = new Sender();

        Scanner scanner = new Scanner(System.in);

        String choice;

        boolean running = true;

In demo we made the objects and called the constructor to access the functions in the sender.

**Constructor made in the classes except Demo:**

**Sender.java**

 public Sender() {

        proContacts = new Contacts[20][2];

        messages = new Message[20];

    }

This the constructor in which both the arrays of the classes **CONTACTS** and **MESSAGE** are initialized.

**Contacts.java**

 public Contacts(String name, String number) {

        this.name = name;

        this.number = number;

    }

This constructor is a **parametrized constructor** which contains both the variables of the Contacts.

**Message.java**

Message(String sender, String reciever, String content, boolean send, boolean seen) {

        this.sender = "Maryam";

        this.reciever = reciever;

        this.content = content;

        this.messageid = String.format("%03d",++counter);

        this.send = send;

        this.seen=seen;

        this.time = LocalDateTime.now();

    }

      //constructor

    public Message( String Reciever, String content) {

        this("Maryam", Reciever, content, false, false);

        this.time=LocalDateTime.now();

    }

There are two constructor one for all the variables but the other one for specific.

**Output:**

**Menu:**

Menu

Choose the function u want to do

1.Add to Contacts

2.Display Contacts

3.Send messages

4.Delete a contact

5.Delete a message

6.Display all messages

7.Display specific messages

8.Display all messages in arranged way

9.Unsend messages

10.Display unseen messages

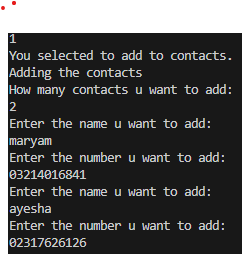
11.Display seen messages

12.Display unsend messages

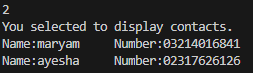
13.Exit the program

Enter the number of your function u want to do(1-12):

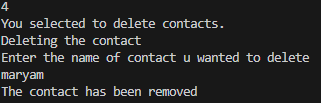
**Add contacts:**

****

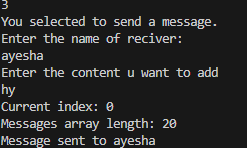
**Display contacts:**

****

**Delete contact:**

****

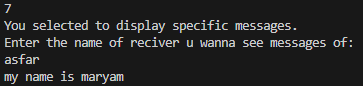
**Send messages:**

****

**Display all messages:**

****

**Display specific message:**

****

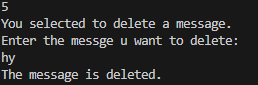
**Display seen messages:**

****

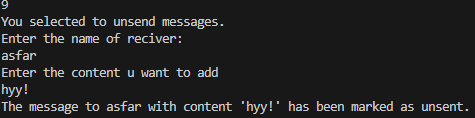
**Display unseen messages:**

****

**Delete a message:**

****

**Unsend messages:**

****

**Display unsend messages:**

****

**Function in Classes:**

**Sender.java**

**Add contacts:**

public void addcontacts() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Adding the contacts");

        System.out.println("How many contacts u want to add:");

        int num = scanner.nextInt();

        scanner.nextLine();

        for (int i = 0; i < num; i++) {

            System.out.println("Enter the name u want to add:");

            String name = scanner.nextLine();

            System.out.println("Enter the number u want to add:");

            String number = scanner.nextLine();

            Contacts contacts = new Contacts(name, number);

            proContacts[countercount][0] = contacts;

            countercount++;

        }

    }

In this we used scanner class which we imported and it is used to take the input from the user

This function is used for add the contacts.

**Delete contact:**

public void deletecontact() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Deleting the contact");

        System.out.println("Enter the name of contact u wanted to delete");

        String name = scanner.nextLine();

        for (int i = 0; i < countercount; i++) {

            if (proContacts[i][0].getName().equals(name)) {

                proContacts[i][0] = null;

                System.out.println("The contact has been removed");

                break;

            }

        }

    }

**Display contacts**

public void displaycontacts() {

        for (int i = 0; i < countercount; i++) {

            if (proContacts[i][0] != null && proContacts[i][0] != null) {

                System.out.println("Name:" + proContacts[i][0].getName() + "\tNumber:" + proContacts[i][0].getNumber());

            } else {

                System.out.println("This contact was deleted or is empty.");

            }

        }

    }

**Send message**

public void sendmessage() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the name of reciver:");

        String name = scanner.nextLine();

        System.out.println("Enter the content u want to add");

        String content = scanner.nextLine();

        boolean contactFound = false;

        for (int i = 0; i < countercount; i++) {

            if (proContacts[i][0] != null && proContacts[i][0].getName().equals(name)) {

                contactFound = true;

                System.out.println("Current index: " + index);

                System.out.println("Messages array length: " + messages.length);

                if (index < messages.length) {

                    Message message = new Message(name, content);

                    messages[index] = message;

                    messages[index].setSend(true);

                    messages[index].setSeen(false);

                    index++;

                    System.out.println("Message sent to " + name);

                } else {

                    System.out.println("Message limit reached. Cannot send more messages.");

                }

                break;

            }

        }

        if (!contactFound) {

            System.out.println("There is no contact with the name: " + name);

        }

    }

**Display messages:**

public void displaymessages() {

        for (int i = 0; i < index; i++) {

            if (messages[i] != null) {

                System.out.println(messages[i].getContent());

            }

        }

    }

**Sending specific messages:**

 public void specificmessages() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the name of reciver u wanna see messages of:");

        String name = scanner.nextLine();

        for (int i = 0; i < countercount; i++) {

            if (proContacts[i][0] != null && proContacts[i][0].getName().equals(name)) {

                for (int j = 0; j < index; j++)

                    if (messages[j] != null && messages[j].getReciever().equals(name)) {

                        messages[j].setSeen(true);

                        System.out.println(messages[j].getContent());

                    }

            }

        }

    }

**Delete messages**

public void deletemessage() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the messge u want to delete:");

        String content = scanner.nextLine();

        for (int i = 0; i < index; i++) {

            if (messages[i].getContent().compareTo(content) == 0) {

                messages[i] = null;

                System.out.println("The message is deleted.");

            }

        }

    }

**Sort message**

 public void sortmessages() {

        for (int i = 0; i < index - 1; i++) {

            for (int j = 0; j < index - 1; j++) {

                if (messages[j].getTime().compareTo(messages[j + 1].getTime()) < 0) {

                    Message temp = messages[j];

                    messages[j] = messages[j + 1];

                    messages[j + 1] = temp;

                }

            }

        }

        for (int i = 0; i < index; i++) {

            if (messages[i] != null) {

                System.out.println(messages[i].getContent());

            } else {

                System.out.println("there are no messages.please send some messages");

            }

        }

    }

In this we sort the message depending on the time the message was sent. The message that was sent at first will come on the top.

**Display seen message**

public void seenmessages() {

        for (int i = 0; i < index; i++) {

            if (messages[i].isSend() == true) {

                if (messages[i].isSeen() == true) {

                    System.out.println(messages[i].getContent());

                }

            }

        }

    }

**Display unseen messages**

public void unseenmessages() {

        for (int i = 0; i < index; i++) {

            if (messages[i].isSend() == true) {

                if (messages[i].isSeen() == false) {

                    System.out.println(messages[i].getContent());

                }

            }

        }

    }

**Unsent the messages**

 public void unsendmessage() {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the name of reciver:");

        String name = scanner.nextLine();

        System.out.println("Enter the content u want to add");

        String content = scanner.nextLine();

        boolean messageFound = false;

        for (int i = 0; i < index; i++) {

            if (messages[i].getReciever().equals(name)) {

                if (messages[i].getContent().compareTo(content) == 0) {

                    messages[i].setSend(false);

                    messages[i] = null;

                    System.out.println(

                            "The message to " + name + " with content '" + content + "' has been marked as unsent.");

                    messageFound = true;

                    break;

                }

            }

        }

        if (!messageFound) {

            System.out.println("Message not found or already unsent.");

        }

    }

**Display the unsend messages:**

public void displayunsend() {

        for (int i = 0; i < index; i++) {

            if (messages[i].isSend() == false) {

                System.out.println(messages[i].getContent());

            }

        }

    }

Demo console based

User driven

do {

            System.out.println("Menu");

            System.out.println("Choose the function u want to do");

            System.out.println("1.Add to Contacts");

            System.out.println("2.Display Contacts");

            System.out.println("3.Send messages");

            System.out.println("4.Delete a contact");

            System.out.println("5.Delete a message");

            System.out.println("6.Display all messages");

            System.out.println("7.Display specific messages");

            System.out.println("8.Display all messages in arranged way");

            System.out.println("9.Unsend messages");

            System.out.println("10.Display unseen messages");

            System.out.println("11.Display seen messages");

            System.out.println("12.Exit the program");

            System.out.println("Enter the number of your function u want to do(1-12):");

            choice = scanner.nextLine();

            switch (choice) {

                case "1":

                    System.out.println("You selected to add to contacts.");

                    send.addcontacts();

                    break;

                case "2":

                    System.out.println("You selected to display contacts.");

                    send.displaycontacts();

                    break;

                case "3":

                    System.out.println("You selected to send a message.");

                    send.sendmessage();

                    break;

                case "4":

                    System.out.println("You selected to delete contacts.");

                    send.deletecontact();

                    break;

                case "5":

                    System.out.println("You selected to delete a message.");

                    send.deletemessage();

                    break;

                case "6":

                    System.out.println("You selected to display all  messages.");

                    send.displaymessages();

                    break;

                case "7":

                    System.out.println("You selected to display specific messages.");

                    send.specificmessages();

                    break;

                case "8":

                    System.out.println("You selected to display the contacts in arranged way.");

                    send.sortmessages();

                    break;

                case "9":

                    System.out.println("You selected to unsend messages.");

                    send.unsendmessage();

                    break;

                case "10":

                    System.out.println("You selected to view unseen messages.");

                    send.unseenmessages();

                    break;

                case "11":

                    System.out.println("You selected to view seen messages.");

                    send.seenmessages();

                    break;

                case "12":

                    System.out.println("You selected to display specific messages.");

                    send.displayunsend();

                    break;

                case "13":

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

                    running = false;

                    break;

                default:

                    System.out.println("Invalid choice! Please select a number between 1 and 12.");

            }

        } while (running);