Mock 24

public static List<Contact> prefill()throws ParseException{

List<Contact> contactList = new ArrayList<>();

SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");

contactList.add(new Contact("John","9876543210","john@gmail.com",sdf.parse("31/01/1990")));

contactList.add(new Contact("Grace","9898798652","grace@gmail.com",sdf.parse("27/02/1987")));

contactList.add(new Contact("Wayne","9787621230","wayne@gmail.com",sdf.parse("15/09/1988")));

contactList.add(new Contact("Joe","9879865898","joe@gmail.com",sdf.parse("23/07/1992")));

contactList.add(new Contact("David","9856985620","david@gmail.com",sdf.parse("24/05/1982")));

contactList.add(new Contact("Jane","9765659820","jane@gmail.com",sdf.parse("13/03/1993")));

contactList.add(new Contact("Choe","9595878580","choe@gmail.com",sdf.parse("26/09/1989")));

contactList.add(new Contact("Chris","9659659790","chris@gmail.com",sdf.parse("15/03/1993")));

return contactList;

}​

**CallLog - Requirement 1**

You have enrolled in a web boot camp. The instructor has given the assignment to implement the CallLog app using Java. Since he has taught Object-oriented programming recently, you have decided to develop the app in OOP approach. The domains given by the instructor to develop the app are Contact, Call, and CallLog. Let's implement a CallLog app by adding features in each requirement.   
  
**Requirement 1:**

Let’s start off by creating two **Contact** objects and check whether they are equal.

1. Create a **Contact** Class with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| mobileNumber | String |
| mailId | String |
| dob | java.util.Date |

1. Mark all the attributes as private
2. Create / Generate appropriate Getters & Setters
3. Add a default constructor and a parameterized constructor to take in all attributes in the given order:   
   **Contact ( String name, String mobileNumber, String mailId, java.util.Date dob )**
4. When the “Contact” object is printed, it should display the following details: **[Override the toString method]**  
   Print format:  
   Name: "name"  
   MobileNumber: "mobileNumber"  
   MailId: "mailId"  
   Dob: "dob"
5. Two Contacts are considered the same if they have the same name, and mobile number. Implement the logic in the appropriate function. (Case – Insensitive) **[Override the equals method]**  
     
   The input format consists of Contact details separated by a comma in the below order,  
   **name, mobileNumber, mailId, dob**

The Input to your program would be details of two Contacts, you need to display their details as given in "5th point(refer above)" and compare the two Contacts and display if the Contacts are same or different.  
  
**Note:**There is an empty line between display statements. Print the empty lines in the main method.

**Sample Input/Output 1:**

Enter contact 1 detail:

**Grace,9898798652,grace@gmail.com,27/02/1987**

Enter contact 2 detail:

**Jane,9765659820,jane@gmail.com,13/03/1993**

Contact 1

Name: Grace

MobileNumber: 9898798652

MailId: grace@gmail.com

Dob: 27/02/1987

Contact 2

Name: Jane

MobileNumber: 9765659820

MailId: jane@gmail.com

Dob: 13/03/1993

Contact 1 and Contact 2 are different

Sample Input/Output 2:

Enter contact 1 detail:

**Wayne,9787621230,wayne@gmail.com,15/09/1988**

Enter contact 2 detail:

**waYNe,9787621230,wayne@gmail.com,15/09/1988**

Contact 1

Name: Wayne

MobileNumber: 9787621230

MailId: wayne@gmail.com

Dob: 15/09/1988

Contact 2

Name: waYNe

MobileNumber: 9787621230

MailId: wayne@gmail.com

Dob: 15/09/1988

Contact 1 is same as Contact 2

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Assign Groups](https://cognizant.e-box.co.in/problem/showTestcaseGroup/8582)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | t1 | Grace,9898798652,grace@gmail.com,27/02/1987 Jane,9765659820,jane@gmail.com,13/03/1993 | Enter contact 1 detail: Enter contact 2 detail:  Contact 1 Name: Grace MobileNumber: 9898798652 MailId: grace@gmail.com Dob: 27/02/1987  Contact 2 Name: Jane MobileNumber: 9765659820 MailId: jane@gmail.com Dob: 13/03/1993  Contact 1 and Contact 2 are different | General |  | | 2 | t3 | Chris,9659659790,chris@gmail.com,15/03/1993 Choe,9595878580,choe@gmail.com,26/09/1989 | Enter contact 1 detail: Enter contact 2 detail:  Contact 1 Name: Chris MobileNumber: 9659659790 MailId: chris@gmail.com Dob: 15/03/1993  Contact 2 Name: Choe MobileNumber: 9595878580 MailId: choe@gmail.com Dob: 26/09/1989  Contact 1 and Contact 2 are different | General |  | | 3 | t4 | David,9856985620,david@gmail.com,24/05/1982 DAVID,9856985620,david@gmail.com,24/05/1982 | Enter contact 1 detail: Enter contact 2 detail:  Contact 1 Name: David MobileNumber: 9856985620 MailId: david@gmail.com Dob: 24/05/1982  Contact 2 Name: DAVID MobileNumber: 9856985620 MailId: david@gmail.com Dob: 24/05/1982  Contact 1 is same as Contact 2 | General |  | | 4 | t2 | Wayne,9787621230,wayne@gmail.com,15/09/1988 waYNe,9787621230,wayne@gmail.com,15/09/1988 | Enter contact 1 detail: Enter contact 2 detail:  Contact 1 Name: Wayne MobileNumber: 9787621230 MailId: wayne@gmail.com Dob: 15/09/1988  Contact 2 Name: waYNe MobileNumber: 9787621230 MailId: wayne@gmail.com Dob: 15/09/1988  Contact 1 is same as Contact 2 | General |  | | 5 | t5 | DavID,9856985620,david@gmail.com,24/05/1982 David,9856985612,david@gmail.com,24/05/1982 | Enter contact 1 detail: Enter contact 2 detail:  Contact 1 Name: DavID MobileNumber: 9856985620 MailId: david@gmail.com Dob: 24/05/1982  Contact 2 Name: David MobileNumber: 9856985612 MailId: david@gmail.com Dob: 24/05/1982  Contact 1 and Contact 2 are different | General |  | | 6 | t6 | Chris,9659659790,chris@gmail.com,15/03/1993 Chris,7899659790,chris@gmail.com,15/03/1993 | Enter contact 1 detail: Enter contact 2 detail:  Contact 1 Name: Chris MobileNumber: 9659659790 MailId: chris@gmail.com Dob: 15/03/1993  Contact 2 Name: Chris MobileNumber: 7899659790 MailId: chris@gmail.com Dob: 15/03/1993  Contact 1 and Contact 2 are different | General |  | |

**CallLog - Requirement 2**

**Requirement 2:**  
   In this requirement, you need to validate the mobile number of the contact.  
  
a)Create a Class **Main** with the following static methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| static Boolean validateMobileNumber(String mobileNumber) | Validate the mobile number based on the rules given below. Returns **true**ifmobileNo is valid else return **false** |

b) While validating mobile number follow the below rules,  
  
1.The mobile number should have prefix as **"+91"**(international calling code).  
2.The mobile number should have 10 digit numbers [0-9].  
3.First digit of the mobile number should be greater than zero.  
  
**Note:** Print "**Mobile number is valid**" if mobile number is valid else print "**Mobile number is invalid**".  
            All the above print statements are present in the main method.  
  
**Sample Input and Output 1:**  
  
Enter the Mobile number to be validated:  
**+919876543210**  
Mobile number is valid  
  
**Sample Input and Output 2:**  
  
Enter the Mobile number to be validated:  
**9652341870**  
Mobile number is invalid

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Assign Groups](https://cognizant.e-box.co.in/problem/showTestcaseGroup/8580)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | 10 | +91000000000 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | | 2 | 8 | 918795462134 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | | 3 | 7 | +9187465123 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | | 4 | 5 | +910912345678 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | | 5 | 6 | +9198765432101 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | | 6 | 1 | +919876543210 | Enter the Mobile number to be validated: Mobile number is valid | General |  | | 7 | 2 | 9652341870 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | | 8 | 4 | +919000000000 | Enter the Mobile number to be validated: Mobile number is valid | General |  | | 9 | 3 | +918876543210 | Enter the Mobile number to be validated: Mobile number is valid | General |  | | 10 | 9 | +908456123789 | Enter the Mobile number to be validated: Mobile number is invalid | General |  | |

**Call Log -Requirement 3**

Now we are gonna compute cost of the calls made based on the callType and duration.  
  
a)Create a class Call with following attributes,

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| contactName | String |
| type | String |
| callType | String |
| cost | Double |
| duration | Date |
| date | Date |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order:**public Call(String contactName, String type, String callType, Date duration, Date date)**  
  
b)Create the following method in the Call class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public void computeCost() | This method computes the cost of the invoking object (based on the below constraints) and stores it in the cost attribute of the object. |

The call details should be given as a comma-separated value in the below order,   
**contactName,type,callType,duration,date**  
  
c) Cost can be calculated by following constraints,

|  |  |  |
| --- | --- | --- |
|  | **Incoming Call Cost** | **Outgoing Call Cost** |
| **Local** | 0.0 | 0.9 |
| **STD** | 0.0 | 1.2 |
| **ISD** | 7.5 | 15.0 |
| **Roaming** | 0.6 | 1.2 |

The above cost are represented as paise per second. Calculate the cost for the whole duration in Rupees.   
  
**Format:**  
Date Format : "**dd-MM-yyyy**"  
Duration Format : "**HH:mm:ss**"   
  
When the “call” object is printed, it should display the following format  
Print format:  
**System.out.printf("%-8s %-8s %-10s %-12s %-10s %s\n","Contact","Type","Call Type","Date","Duration","Cost");**  
The cost should be printed with 2 decimal digits.   
  
**Sample Input and Output:**  
  
Enter the number of calls:  
**5  
John,Incoming,STD,00:01:50,02-05-2018  
Grace,Incoming,ISD,00:10:55,28-07-2018  
Wayne,Incoming,Local,00:05:35,10-07-2017  
Joe,Incoming,STD,00:01:50,30-06-2018  
David,Outgoing,Local,00:10:15,22-05-2018**  
Contact    Type          Call Type         Date           Duration     Cost  
John        Incoming      STD        02-05-2018      00:01:50      0.00  
Grace     Incoming       ISD        28-07-2018      00:10:55      49.13  
Wayne   Incoming     Local      10-07-2017      00:05:35      0.00  
Joe          Incoming      STD        30-06-2018      00:01:50      0.00  
David     Outgoing     Local      22-05-2018      00:10:15      5.54

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Assign Groups](https://cognizant.e-box.co.in/problem/showTestcaseGroup/8584)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | s2 | 5 Joe,Incoming,STD,00:01:50,30-06-2018 David,Outgoing,Local,00:10:15,22-05-2018 John,Outgoing,STD,00:20:25,01-01-2018 Jane,Outgoing,ISD,00:15:20,03-08-2017 Choe,Outgoing,STD,00:30:03,31-05-2018 | Enter the number of calls: Contact  Type     Call Type  Date         Duration   Cost Joe      Incoming STD        30-06-2018   00:01:50   0.00 David    Outgoing Local      22-05-2018   00:10:15   5.54 John     Outgoing STD        01-01-2018   00:20:25   14.70 Jane     Outgoing ISD        03-08-2017   00:15:20   138.00 Choe     Outgoing STD        31-05-2018   00:30:03   21.64 | General |  | | 2 | s4 | 3 Wayne,Incoming,Local,00:05:35,10-07-2017 Joe,Incoming,STD,00:01:50,30-06-2018 John,Incoming,STD,00:01:50,02-05-2018 | Enter the number of calls: Contact  Type     Call Type  Date         Duration   Cost Wayne    Incoming Local      10-07-2017   00:05:35   0.00 Joe      Incoming STD        30-06-2018   00:01:50   0.00 John     Incoming STD        02-05-2018   00:01:50   0.00 | General |  | | 3 | s1 | 5 John,Incoming,STD,00:01:50,02-05-2018 Grace,Incoming,ISD,00:10:55,28-07-2018 Wayne,Incoming,Local,00:05:35,10-07-2017 Joe,Incoming,STD,00:01:50,30-06-2018 David,Outgoing,Local,00:10:15,22-05-2018 | Enter the number of calls: Contact  Type     Call Type  Date         Duration   Cost John     Incoming STD        02-05-2018   00:01:50   0.00 Grace    Incoming ISD        28-07-2018   00:10:55   49.13 Wayne    Incoming Local      10-07-2017   00:05:35   0.00 Joe      Incoming STD        30-06-2018   00:01:50   0.00 David    Outgoing Local      22-05-2018   00:10:15   5.54 | General |  | | 4 | s5 | 7 Grace,Incoming,ISD,00:10:55,28-07-2018 Wayne,Incoming,Local,00:05:35,10-07-2017 Joe,Incoming,Roaming,00:01:50,30-06-2018 David,Outgoing,Local,00:10:15,22-05-2018 John,Outgoing,STD,00:20:25,01-01-2018 Jane,Outgoing,ISD,00:15:20,03-08-2017 Choe,Outgoing,Roaming,00:30:03,31-05-2018 | Enter the number of calls: Contact  Type     Call Type  Date         Duration   Cost Grace    Incoming ISD        28-07-2018   00:10:55   49.13 Wayne    Incoming Local      10-07-2017   00:05:35   0.00 Joe      Incoming Roaming    30-06-2018   00:01:50   0.66 David    Outgoing Local      22-05-2018   00:10:15   5.54 John     Outgoing STD        01-01-2018   00:20:25   14.70 Jane     Outgoing ISD        03-08-2017   00:15:20   138.00 Choe     Outgoing Roaming    31-05-2018   00:30:03   21.64 | General |  | | 5 | s3 | 6 Joe,Incoming,STD,00:01:50,30-06-2018 David,Outgoing,Local,00:10:15,22-05-2018 John,Outgoing,STD,00:20:25,01-01-2018 Jane,Outgoing,ISD,00:15:20,03-08-2017 Choe,Outgoing,STD,00:30:03,31-05-2018 Chris,Outgoing,Local,00:04:30,21-06-2018 | Enter the number of calls: Contact  Type     Call Type  Date         Duration   Cost Joe      Incoming STD        30-06-2018   00:01:50   0.00 David    Outgoing Local      22-05-2018   00:10:15   5.54 John     Outgoing STD        01-01-2018   00:20:25   14.70 Jane     Outgoing ISD        03-08-2017   00:15:20   138.00 Choe     Outgoing STD        31-05-2018   00:30:03   21.64 Chris    Outgoing Local      21-06-2018   00:04:30   2.43 | General |  |   **CallLog - Requirement 4**  **Requirement 4:** In this requirement develop a feature in which you can search a List of Calls by date, and contact.  a) Create a Class Contact with the following attributes:   |  |  | | --- | --- | | **Member Field Name** | **Type** | | name | String | | mobileNumber | String | | mailId | String | | dob | java.util.Date |   The following methods are present in the Contact class   |  |  | | --- | --- | | **Method Name** | **Method Description** | | static List<Contact> prefill() | This method returns the List of prefilled Contact objects. |   Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Contact( String name, String mobileNumber, String mailId, java.util.Date dob )**   b) Create a Class Call with the following attributes:    |  |  | | --- | --- | | **Member Field Name** | **Type** | | type | String | | callType | String | | cost | Double | | duration | java.util.Date | | date | java.util.Date | | contact | Contact |   Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Call( String type, String callType, Double cost, java.util.Date duration, java.util.Date date, Contact contact )**  c) Create a class **CallBO** with the following methods,   |  |  | | --- | --- | | **Method Name** | **Description** | | public List<Call> findCall(List<Call> callList,java.util.Date date) | This method accepts a list of calls and date as arguments and returns a list of calls that matches with the given date. | | public List<Call> findCall(List<Call> callList,String contactName) | This method accepts a list of calls and contact name as arguments and returns a list of calls that matches with the given contact name(case -insensitive). |   The call details should be given as a comma-separated value in the below order, type, callType, cost, duration, date, contact's name **Format:** Date Format : "**dd/MM/yyyy**" Duration Format : "**HH:mm:ss**"  Print format: System.out.format("%-13s %-15s %-10s %-10s %-10s %-15s %s\n","Name","Mobile Number","Type", "Call Type","Cost" ,"Duration","Date");  **Note:**The call lists are displayed in the main method.             If any other choice is selected, display "**Invalid choice**"             If the search detail is not found, display "**No calls from <contactName>**"for search using contactName and             "**No calls on <date>**" for search using date,             Display two digit after the decimal point for Double Datatype.  **Sample Input & Output 1:**  Enter the number of calls: **6 Incoming,STD,0.25,00:01:50,20/06/2018,John Incoming,ISD,5.20,00:10:55,15/07/2018,Grace Outgoing,Local,2.00,00:10:15,10/02/2018,David Incoming,Local,0.00,00:05:35,10/07/2018,John Outgoing,STD,5.05,00:20:25,02/05/2018,John Incoming,STD,10.00,00:01:50,18/04/2018,Joe** Enter a search type: 1.By a name 2.By a call date **1** Enter the name: **john** Name          Mobile Number   Type              Call Type  Cost       Duration        Date John             9876543210           Incoming   STD               0.25       00:01:50         20/06/2018 John             9876543210           Incoming   Local            0.00       00:05:35         10/07/2018 John             9876543210           Outgoing   STD               5.05       00:20:25         02/05/2018  **Sample Input & Output 2:**  Enter the number of calls: **4 Incoming,STD,0.25,00:01:50,20/06/2018,John Incoming,ISD,5.20,00:10:55,15/07/2018,Grace Outgoing,STD,5.05,00:20:25,20/06/2018,Jane Outgoing,Local,1.50,00:04:30,01/08/2018,Chris** Enter a search type: 1.By a name 2.By a call date **2** Enter the call date: **20/06/2018** Name          Mobile Number   Type             Call Type  Cost       Duration        Date John             9876543210           Incoming   STD             0.25        00:01:50         20/06/2018 Jane            9765659820           Outgoing   STD             5.05        00:20:25         20/06/2018  **Test Cases**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | [Assign Groups](https://cognizant.e-box.co.in/problem/showTestcaseGroup/8581)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | 1 | 6 Incoming,STD,0.25,00:01:50,20/06/2018,John Incoming,ISD,5.20,00:10:55,15/07/2018,Grace Outgoing,Local,2.00,00:10:15,10/02/2018,David Incoming,Local,0.00,00:05:35,10/07/2018,John Outgoing,STD,5.05,00:20:25,02/05/2018,John Incoming,STD,10.00,00:01:50,18/04/2018,Joe 1 john | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the name: Name          Mobile Number   Type       Call Type  Cost       Duration        Date John          9876543210      Incoming   STD        0.25       00:01:50        20/06/2018 John          9876543210      Incoming   Local      0.00       00:05:35        10/07/2018 John          9876543210      Outgoing   STD        5.05       00:20:25        02/05/2018 | General |  | | 2 | 6 | 0 1 john | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the name: No calls from john | General |  | | 3 | 5 | 4 Incoming,STD,10.00,00:01:50,18/04/2018,Joe Outgoing,ISD,55.75,00:15:20,20/06/2018,Jane Outgoing,STD,11.00,00:30:03,09/08/2018,Choe Incoming,STD,10.00,00:01:50,10/07/2018,Joe 0 | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Invalid choice | General |  | | 4 | 8 | 4 Outgoing,STD,5.05,00:20:25,02/05/2018,John Incoming,STD,0.25,00:01:50,20/06/2018,Wayne Outgoing,Local,2.00,00:10:15,10/07/2018,David Incoming,ISD,5.20,00:10:55,15/07/2018,Wayne 2 20/05/2018 | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the call date: No calls on 20/05/2018 | General |  | | 5 | 7 | 5 Incoming,ISD,5.20,00:10:55,15/07/2018,Grace Incoming,Local,0.00,00:05:35,10/07/2018,Wayne Outgoing,Local,2.00,00:10:15,10/07/2018,David Outgoing,STD,5.05,00:20:25,02/05/2018,John Incoming,STD,10.00,00:01:50,18/04/2018,Joe 1 choe | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the name: No calls from choe | General |  | | 6 | 3 | 10 Incoming,STD,0.25,00:01:50,20/06/2018,John Outgoing,Local,1.50,00:04:30,10/07/2018,Chris Incoming,ISD,5.20,00:10:55,15/07/2018,Grace Incoming,Local,0.00,00:05:35,10/07/2018,Wayne Outgoing,Local,2.00,00:10:15,10/07/2018,David Outgoing,STD,5.05,00:20:25,02/05/2018,John Incoming,STD,10.00,00:01:50,18/04/2018,Joe Outgoing,ISD,55.75,00:15:20,20/06/2018,Jane Outgoing,STD,11.00,00:30:03,09/08/2018,Choe Incoming,STD,10.00,00:01:50,10/07/2018,Joe 2 10/07/2018 | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the call date: Name          Mobile Number   Type       Call Type  Cost       Duration        Date Chris         9659659790      Outgoing   Local      1.50       00:04:30        10/07/2018 Wayne         9787621230      Incoming   Local      0.00       00:05:35        10/07/2018 David         9856985620      Outgoing   Local      2.00       00:10:15        10/07/2018 Joe           9879865898      Incoming   STD        10.00      00:01:50        10/07/2018 | General |  | | 7 | 2 | 4 Incoming,STD,0.25,00:01:50,20/06/2018,John Incoming,ISD,5.20,00:10:55,15/07/2018,Grace Outgoing,STD,5.05,00:20:25,20/06/2018,Jane Outgoing,Local,1.50,00:04:30,01/08/2018,Chris 2 20/06/2018 | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the call date: Name          Mobile Number   Type       Call Type  Cost       Duration        Date John          9876543210      Incoming   STD        0.25       00:01:50        20/06/2018 Jane          9765659820      Outgoing   STD        5.05       00:20:25        20/06/2018 | General |  | | 8 | 4 | 10 Incoming,STD,0.25,00:01:50,20/06/2018,Wayne Outgoing,Local,2.00,00:10:15,10/07/2018,David Outgoing,STD,5.05,00:20:25,02/05/2018,John Incoming,ISD,5.20,00:10:55,15/07/2018,Wayne Incoming,Local,0.00,00:05:35,10/07/2018,Wayne Incoming,STD,10.00,00:01:50,18/04/2018,Joe Outgoing,ISD,55.75,00:15:20,20/06/2018,Jane Outgoing,STD,11.00,00:30:03,09/08/2018,Choe Outgoing,Local,1.50,00:04:30,10/07/2018,Wayne Incoming,STD,10.00,00:01:50,10/07/2018,Joe 1 Wayne | Enter the number of calls: Enter a search type: 1.By a name 2.By a call date Enter the name: Name          Mobile Number   Type       Call Type  Cost       Duration        Date Wayne         9787621230      Incoming   STD        0.25       00:01:50        20/06/2018 Wayne         9787621230      Incoming   ISD        5.20       00:10:55        15/07/2018 Wayne         9787621230      Incoming   Local      0.00       00:05:35        10/07/2018 Wayne         9787621230      Outgoing   Local      1.50       00:04:30        10/07/2018 | General |  | |   **Call Log -Requirement 5**  Now in this requirement, we have to display a month wise report of the call history.   a)Create a class Call with following attributes,   |  |  | | --- | --- | | **Member Field Name** | **Type** | | contact | Contact | | type | String | | callType | String | | cost | Double | | duration | Date | | date | Date |   Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order:**public Call(Contact contact, String type, String callType, Double cost, Date duration, Date date)**   b) Create a class Contact with the following attributes,   |  |  | | --- | --- | | **Member Field Name** | **Type** | | name | String | | mobileNumber | String | | mailId | String | | dob | Date |   Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order:**public Contact(String name, String mobileNumber, String mailId, Date dob)**  c) Create the following method in the Call class,   |  |  | | --- | --- | | **Method Name** | **Description** | | public static Map<String,Integer> monthWiseCount(List<Call> callList) | This method takes list of calls asargument and return a tree map which has month name (in 3 letter format) as key and number of calls made in the month as value. |   d) Create the following method in the Contact class,   |  |  | | --- | --- | | **Method Name** | **Description** | | public static List<Contact> preFill() | This method return a list of prefilled Contacts, this will be given in the template |   The call details should be given as a comma-separated value in the below order, **contactName,type,cost,allType,duration,date**   Print format: System.out.printf("%-10s %s\n","Month", "Count");   **Sample Input and Output:**  Enter the number of calls: **5 Wayne,Incoming,Local,0,00:05:35,10-07-2018 Joe,Incoming,STD,10.00,00:01:50,30-06-2018 David,Outgoing,Local,2.00,00:10:15,22-05-2018 John,Outgoing,STD,5.05,00:20:25,01-01-2018 Jane,Outgoing,ISD,55.75,00:15:20,03-08-2018** Month      Count Aug        1 Jan        1 Jul        1 Jun        1 May        1  **Test Cases**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | [Assign Groups](https://cognizant.e-box.co.in/problem/showTestcaseGroup/8585)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | s4 | 9 John,Incoming,STD,0.25,00:01:50,02-05-2018 Grace,Incoming,ISD,5.20,00:10:55,28-07-2018 Wayne,Incoming,Local,0,00:05:35,10-07-2018 Joe,Incoming,STD,10.00,00:01:50,30-06-2018 David,Outgoing,Local,2.00,00:10:15,22-05-2018 John,Outgoing,STD,5.05,00:20:25,01-01-2018 Jane,Outgoing,ISD,55.75,00:15:20,03-08-2018 Choe,Outgoing,STD,11.00,00:30:03,31-05-2018 Chris,Outgoing,Local,1.50,00:04:30,21-06-2018 | Enter the number of calls: Month      Count Aug        1 Jan        1 Jul        2 Jun        2 May        3 | General |  | | 2 | s3 | 5 David,Outgoing,Local,2.00,00:10:15,22-05-2018 John,Outgoing,STD,5.05,00:20:25,01-01-2018 Jane,Outgoing,ISD,55.75,00:15:20,03-08-2018 Choe,Outgoing,STD,11.00,00:30:03,31-05-2018 Chris,Outgoing,Local,1.50,00:04:30,21-06-2018 | Enter the number of calls: Month      Count Aug        1 Jan        1 Jun        1 May        2 | General |  | | 3 | s1 | 5 Wayne,Incoming,Local,0,00:05:35,10-07-2018 Joe,Incoming,STD,10.00,00:01:50,30-06-2018 David,Outgoing,Local,2.00,00:10:15,22-05-2018 John,Outgoing,STD,5.05,00:20:25,01-01-2018 Jane,Outgoing,ISD,55.75,00:15:20,03-08-2018 | Enter the number of calls: Month      Count Aug        1 Jan        1 Jul        1 Jun        1 May        1 | General |  | | 4 | s2 | 5 John,Incoming,STD,0.25,00:01:50,02-05-2018 Grace,Incoming,ISD,5.20,00:10:55,28-07-2018 Wayne,Incoming,Local,0,00:05:35,10-07-2018 Joe,Incoming,STD,10.00,00:01:50,30-06-2018 David,Outgoing,Local,2.00,00:10:15,22-05-2018 | Enter the number of calls: Month      Count Jul        2 Jun        1 May        2 | General |  | |   **CallLog - Requirement 6**  **Requirement 6:** In this requirement, Let's develop a feature to find the favourite contact from the Call Log based on total number of calls from and to the user.  Create a Class Contact with the following private attributes   |  |  | | --- | --- | | **Attributes** | **Datatype** | | name | String | | mobileNumber | String | | mailId | String | | dob | Date |   Create / Generate appropriate Getters & Setters Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Contact ( String name, String mobileNumber, String mailId, java.util.Date dob )**  The following methods are present in the Contact class   |  |  | | --- | --- | | **Method Name** | **Method Description** | | static List<Contact> prefill() | This method returns the List of prefilled Contact objects. |   Create a Class Call with private attributes   |  |  | | --- | --- | | **Attributes** | **Datatype** | | type | String | | callType | String | | cost | Double | | duration | Date | | date | Date | | contact | Contact |   Create / Generate appropriate Getters & Setters Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Call(String type, String callType, Double cost, Date duration,Date date, Contact contact)**  Create a Class CallLog with private attributes   |  |  | | --- | --- | | **Attributes** | **Datatype** | | callList | List<Call> |   Create / Generate appropriate Getters & Setters Add a default constructor and a parameterized constructor to take in all attributes in the given order: **CallLog(List<Call> callList)**  The following methods are present in the CallLog class   |  |  | | --- | --- | | **Method Name** | **Method Description** | | Contact getFavouriteContact() | This method returns a Contact that has maximum number of calls (both incoming and outgoing) to the user from list of Call objects in the invoking object. |   The call inputs are given in comma separated format in the below order, **type, callType, cost, duration, date, contactName**  Note : While creating CallLog object, create a empty list for callList attribute.            After calling appropriate methods, display the name of favourite contact in the main method.                  Use "dd/MM/yyyy" for date attribute.            Use "HH:mm:ss" for duration attribute.  **Sample Input/Output:**    Enter the number of calls:  **6**  **Incoming,STD,0.25,00:01:50,22/01/2018,John**  **Incoming,ISD,5.20,00:10:55,15/02/2018,Grace**  **Incoming,Local,0,00:05:35,05/02/2018,Wayne**  **Outgoing,STD,11.00,00:30:03,01/07/2018,John**  **Outgoing,Local,1.50,00:04:30,02/12/2017,John**  **Incoming,STD,10.00,00:01:50,02/05/2018,Joe**  The favourite contact is John  **Test Cases**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | [Assign Groups](https://cognizant.e-box.co.in/problem/showTestcaseGroup/8583)   | **SNo** | **Name** | **Input** | **Output** | **Purpose** | **Sample** | | --- | --- | --- | --- | --- | --- | | 1 | t4 | 6 Incoming,Local,0,00:05:35,05/02/2018,Wayne Incoming,STD,10.00,00:01:50,02/05/2018,Joe Outgoing,Local,2.00,00:10:15,03/05/2018,David Incoming,ISD,5.20,00:10:55,15/02/2018,David Incoming,STD,0.25,00:01:50,22/01/2018,David Outgoing,Local,1.50,00:04:30,02/12/2017,Chris | Enter the number of calls: The favourite contact is David | General |  | | 2 | t2 | 7 Incoming,STD,0.25,00:01:50,22/01/2018,John Incoming,ISD,5.20,00:10:55,15/02/2018,Grace Incoming,Local,0,00:05:35,05/02/2018,John Incoming,STD,10.00,00:01:50,02/05/2018,Joe Outgoing,Local,2.00,00:10:15,03/05/2018,Joe Outgoing,STD,11.00,00:30:03,01/07/2018,Choe Outgoing,Local,1.50,00:04:30,02/12/2017,Joe | Enter the number of calls: The favourite contact is Joe | General |  | | 3 | t5 | 7 Incoming,STD,0.25,00:01:50,22/01/2018,John Incoming,ISD,5.20,00:10:55,15/02/2018,Grace Incoming,Local,0,00:05:35,05/02/2018,Wayne Incoming,STD,10.00,00:01:50,02/05/2018,Joe Outgoing,STD,5.05,00:20:25,28/02/2018,Wayne Outgoing,Local,1.50,00:04:30,02/12/2017,Chris Outgoing,STD,11.00,00:30:03,01/07/2018,Choe | Enter the number of calls: The favourite contact is Wayne | General |  | | 4 | t1 | 6 Incoming,STD,0.25,00:01:50,22/01/2018,John Incoming,ISD,5.20,00:10:55,15/02/2018,Grace Incoming,Local,0,00:05:35,05/02/2018,Wayne Outgoing,STD,11.00,00:30:03,01/07/2018,John Outgoing,Local,1.50,00:04:30,02/12/2017,John Incoming,STD,10.00,00:01:50,02/05/2018,Joe | Enter the number of calls: The favourite contact is John | General |  | | 5 | t3 | 5 Incoming,ISD,5.20,00:10:55,15/02/2018,Grace Outgoing,ISD,55.75,00:15:20,30/01/2018,Grace Outgoing,STD,11.00,00:30:03,01/07/2018,Grace Incoming,Local,0,00:05:35,05/02/2018,Grace Incoming,STD,0.25,00:01:50,22/01/2018,John | Enter the number of calls: The favourite contact is Grace | General |  | | |