BookABikeService-Infosys ...

0%

SecondLargestIsPalind

Problem Statement

This question aims to test your knowledge of Regex.
Implement as per the given requirement.

Requirement:

Tuneup, a motor vehicle service performs a series of maintenance procedures carried out at a specific time interval or after the vehicle has traveled a certain distance.

Over time your bike will go through a lot of wear and tear. Sometimes it's only when we come face to face with a mechanical problem that we realize how fragile a bike can be. Even small failures can make a huge impact on the functionality of the bike. Little problems such as moved brake pads can subtly cause your braking power to decrease. Aim to get your bike checked over every few months.

So, to save time, the Tuneup service needs to be automated for which a series of validations are required for the below components:

a. customerName

b. bikeNo

c. phoneNo

d. serviceType

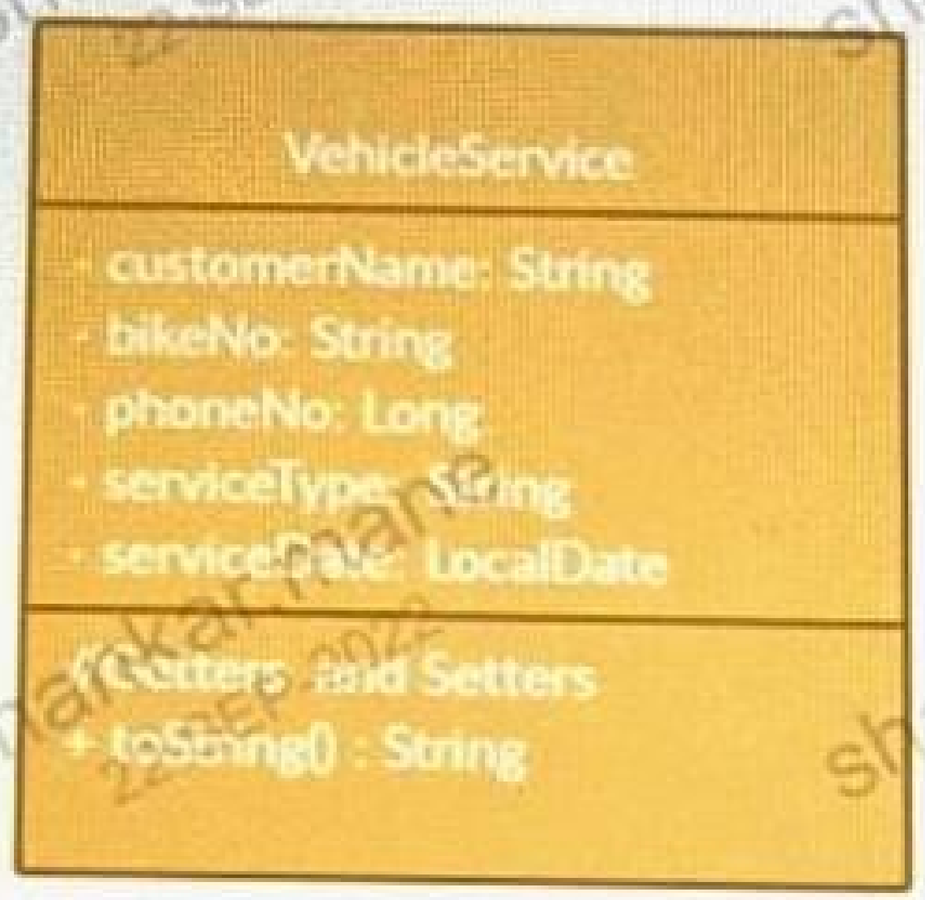
e. ServiceDate

Class Diagram:

BookABikeService-Infosys ... 0%

SecondLargestIsPali

Class Diagram:



Instructions:

As per the above requirement, the **VehicleService** class which is a simple model class, and the **Main** class accepting all the above parameters with the proper function calls are already provided to you.

1. Implement the below methods in the **Validator** class to complete the validation process:
    1. **public static Boolean isValidServiceType (String serviceType):**
        1. This method validates the **serviceType**.
        2. The **service type** should be either **General** or **Express**.
        3. If the above condition is satisfied, return true, otherwise, return false.
2. **public static Boolean isValidPhoneNo(Long phoneNo):**
    1. This method validates phoneNo

1. Implement the below methods in the **Validator** class to complete the validation process:
    1. **public static Boolean isValidServiceType (String serviceType):**
        1. This method validates the **serviceType**.
        2. The **service type** should be either **General** or **Express**.
        3. If the above condition is satisfied, return true, otherwise, return false.
    2. **public static Boolean isValidPhoneNo(Long phoneNo):**
        1. This method validates **phoneNo**.
        2. PhoneNo must be a 10-digit number and all the numbers should not be the same.
        3. If the above condition is satisfied, return true, otherwise, return false.
        4. Example-
            - Valid-**7416306445,8179845754**
            - Invalid- **741630644,9999999999**
    3. **public static Boolean isValidCustomerName(String customerName):**
        1. This method validates the received **customerName.**
        2. It should **NOT** be empty or just spaces.
        3. It must contain alphabets only.
        4. It should have a **minimum of one character**.
        5. It shouldn't have more than one word.
        6. Each word should start with an **uppercase** alphabet. All the other characters should be lowercase alphabets.
        7. If the above conditions are satisfied, return true, otherwise, return false.
        8. Example-
            - Valid-**Sachin, John, J**
            - Invalid- **Sac@hin, Sachin2, 234567, j**
    4. **public static Boolean isValidBikeNo (String bikeNo):**
        1. This method validates **bikeNo.**
        2. It must contain **2 uppercase characters** followed by **2 digits** followed by **2**

6. Each word should start with an **uppercase** alphabet. All the other characters should be lowercase alphabets.
7. If the above conditions are satisfied, return true, otherwise, return false.
8. Example-
    - Valid-**Sachin, John, J**
    - Invalid- **Sac@hin, Sachin2, 234567, j**

4. **public static Boolean isValidBikeNo (String bikeNo):**
    1. This method validates **bikeNo.**
    2. It must contain **2 uppercase characters** followed by **2 digits** followed by **2 uppercase characters** followed by **4 digits.**
    3. If the above conditions are satisfied, return true, otherwise, return false.
    4. Example-
        - Valid-**AP31CW3008, KA12CA1991, AP11CW3008, AP00CW3333, AP13CW0000**
        - Invalid- **aP31CW3008, AP009W3008, AP31CWH000, AP31cW3008**
5. **public static Boolean isValidServiceDate (LocalDate lastServiceDate):**
    1. This method validates the **lastServiceDate.**
    2. The last service date shouldn't be a future date.
    3. If the above condition is satisfied, return true, otherwise, return false.
    4. Example- If Today's date is **23rd June 2021**
        - Valid-**29th August 2018, 1st January 2019.**
        - Invalid-**24th June 2021,23rd June 2021, 2nd March 2030, 26th May 3030.**

Fully implemented **VehicleService.java**, **Main.java,** and partially implemented **Validator.java are** already given to you.

Note:

- Valid-29th August 2018, 1st January 2019.
- Invalid-24th June 2021,23rd June 2021, 2nd March 2030, 26th May 3030.

Fully implemented **VehicleService.java**, **Main.java**, and partially implemented **Validator.java are** already given to you.

Note:

- Please don't alter/change the codes which have already been provided.
- Whatever class/interface you are adding OR already provided should not be 'public'.

Sample Input	Sample Output	Explanation
Srija AP31CW0896 8179892186 General 25/06/2020	true true true true true	All the details provided are valid.
Srija2 AP31CW1841 8179892186 General 25/06/2020	false true true true true	The customerName is invalid.
Srija	true	The bikeNo is invalid.

25/06/2020	true	
Srija AP31CW18411 8179892186 General 25/06/2020	true false true true true	The bikeNo is invalid.
Srija AP31CW1841 81798921 General 25/06/2020	true true false true true	The phoneNo is invalid.
Srija AP31CW1841 8179892186 Gold 25/06/2020	true true true false true	The serviceType is invalid.
Srija AP31CW1841 8179892186 General 26/11/3030	true true true true false	The lastServiceDate is invalid.

Modifiers.java | Employee.java | Course.java | *Practice.java | DemoApplication.java

```java
package practicePackage;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class Practice {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        VehicleService vehicleService=new VehicleService();
        String customerName =sc.nextLine();
        vehicleService.setCustomerName(customerName);
        String bikeNo =sc.nextLine();
        vehicleService.setBikeNo(bikeNo);
        Long phoneNo=sc.nextLong();
        vehicleService.setPhoneNo(phoneNo);
        String serviceType=sc.next();
        vehicleService.setServiceType(serviceType);
        String serviceDateInString =sc.next();
        DateTimeFormatter formattor =DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate serviceDate =LocalDate.parse(serviceDateInString,formattor);
        vehicleService.setServiceDate(serviceDate);
        Validator.validate(vehicleService);
        sc.close();
    }
}
class VehicleService{
    private String customerName;
    private String bikeNo;
    private Long phoneNo;
    private String serviceType;
    private LocalDate serviceDate;
```

Modifiers.java | Employee.java | Course.java | *Practice.java × | DemoApplication.java

```
33      public String getCustomerName() {
34          return customerName;
35      }
36      public void setCustomerName(String customerName) {
37          this.customerName = customerName;
38      }
39      public String getBikeNo() {
40          return bikeNo;
41      }
42      public void setBikeNo(String bikeNo) {
43          this.bikeNo = bikeNo;
44      }
45      public Long getPhoneNo() {
46          return phoneNo;
47      }
48      public void setPhoneNo(Long phoneNo) {
49          this.phoneNo = phoneNo;
50      }
51      public String getServiceType() {
52          return serviceType;
53      }
54      public void setServiceType(String serviceType) {
55          this.serviceType = serviceType;
56      }
57      public LocalDate getServiceDate() {
58          return serviceDate;
59      }
60      public void setServiceDate(LocalDate serviceDate) {
61          this.serviceDate = serviceDate;
62      }
63      @Override
64      public String toString() {
```

Modifiers.java | Employee.java | Course.java | *Practice.java | DemoApplication.java

```
    @Override
    public String toString() {
        return "VehicleService [customerName=" + customerName + ", bikeNo=" + bikeNo + ", phoneNo=" + phoneNo
                + ", serviceType=" + serviceType + ", serviceDate=" + serviceDate + "]";
    }

}
class Validator{
    public static void validate(VehicleService vehicleService) {
        System.out.println(isValidCustomerName(vehicleService.getCustomerName()));
        System.out.println(isValidBikeNo(vehicleService.getBikeNo()));
        System.out.println(isValidPhoneNo(vehicleService.getPhoneNo()));
        System.out.println(isValidServiceType(vehicleService.getServiceType()));
        System.out.println(isValidServiceDate(vehicleService.getServiceDate()));
    }
    public static Boolean isValidCustomerName(String customerName) {
        //String regexp="[A-Z]{1}[a-z]{0,}";
        String regexp="[A-Z][a-z]*";
        if(customerName.matches(regexp)) {
            return true;
        }
        return false;

    }
    public static Boolean isValidBikeNo(String bikeNo) {
        String regexp ="[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}";
        if(bikeNo.matches(regexp)) {
            return true;
        }
        return false;

    }
```

Modifiers.java | Employee.java | Course.java | *Practice.java | DemoApplication.java

```
101
102     public static Boolean isValidBikeNo(String bikeNo) {
103         String regexp = "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}";
104         if (bikeNo.matches(regexp)) {
105             return true;
106         }
107         return false;
108
109     }
110     public static Boolean isValidPhoneNo(Long phoneNo) {
111       if(phoneNo>=1000000000 && phoneNo<9999999999L && phoneNo%1111111111!=0 )  {
112         return true;
113         }
114         return false;
115
116     }
117
118 //  public static Boolean isValidPhoneNo(Long phoneNo) {
119 //      String regexp = "[0-9]{10}";
120 //      String phone = phoneNo.toString();
121 //      if (phone.matches(regexp) && !((phoneNo % 1111111111) == 0)) {
122 //          return true;
123 //      }
124 //      return false;
125 //  }
126
127     public static Boolean isValidServiceType(String serviceType) {
128         if (serviceType.equals("General") || serviceType.equals("Express")) {
129             return true;
130         }
131         return false;
132     }
```

Modifiers.java | Employee.java | Course.java | *Practice.java × | DemoApplication.java

```
123         }
124         return false;
125     }
126
127     public static Boolean isValidServiceType(String serviceType) {
128         if (serviceType.equals("General") || serviceType.equals("Express")) {
129             return true;
130         }
131         return false;
132     }
133
134     public static Boolean isValidServiceDate(LocalDate lastServiceDate) {
135         if (lastServiceDate.isAfter(LocalDate.now())) {
136             return false;
137         }
138         return true;
139     }
140
141 }
142
```

Error Log | Declaration | Console × | JUnit | Javadoc

<terminated> Practice [Java Application] D:\TypeScript SDK 4.5.5\SoftwareCenterApplications\Eclipse IDE for Java Developers\e

```
Srija
AP31CW0896
8179892186
General
25/06/2022
true
true
true
true
true
```

```
146
147 //the current date today is 22/09/2022
148
```

Error Log | Declaration | Console × | JUnit | Javadoc

<terminated> Practice [Java Application] D:\TypeScript SDK 4.5.5\SoftwareCenterApplic

```
Srija2
AP31CW18411
81798921
Gold
23/09/2022
false
false
false
false
false
```

```
146
147 //the current date today is 22/09/2022
148
```

Error Log | Declaration | Console | JUnit | Javadoc

<terminated> Practice [Java Application] D:\TypeScript SDK 4.5.5\SoftwareCenterApp

```
Srija2
AP31CW18411
81798921
Gold
22/09/2022
false
false
false
false
true
```