Problem Statement

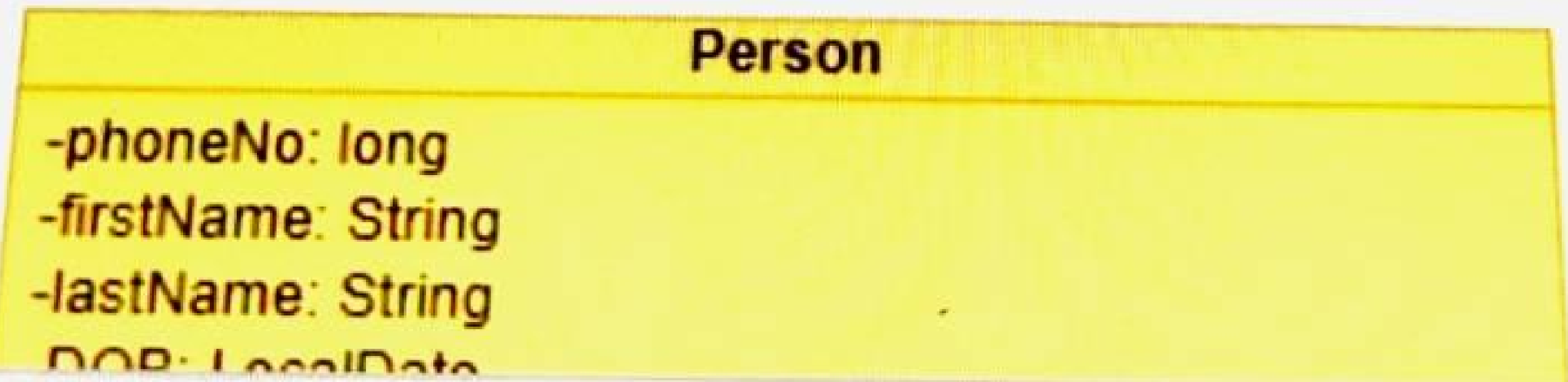
This question aims to test your knowledge of Collections, String, Wrapper class, and LocalDate functions. Implement as per the given requirement.

A leading MNC in India wants to develop an application that will automate the process of generating a Learner License for few states. With help of this process a Person can get the Learner's License details after applying with his/her phone number, first name, last name, DOB, state, and district. So, implement the below requirement as provided.

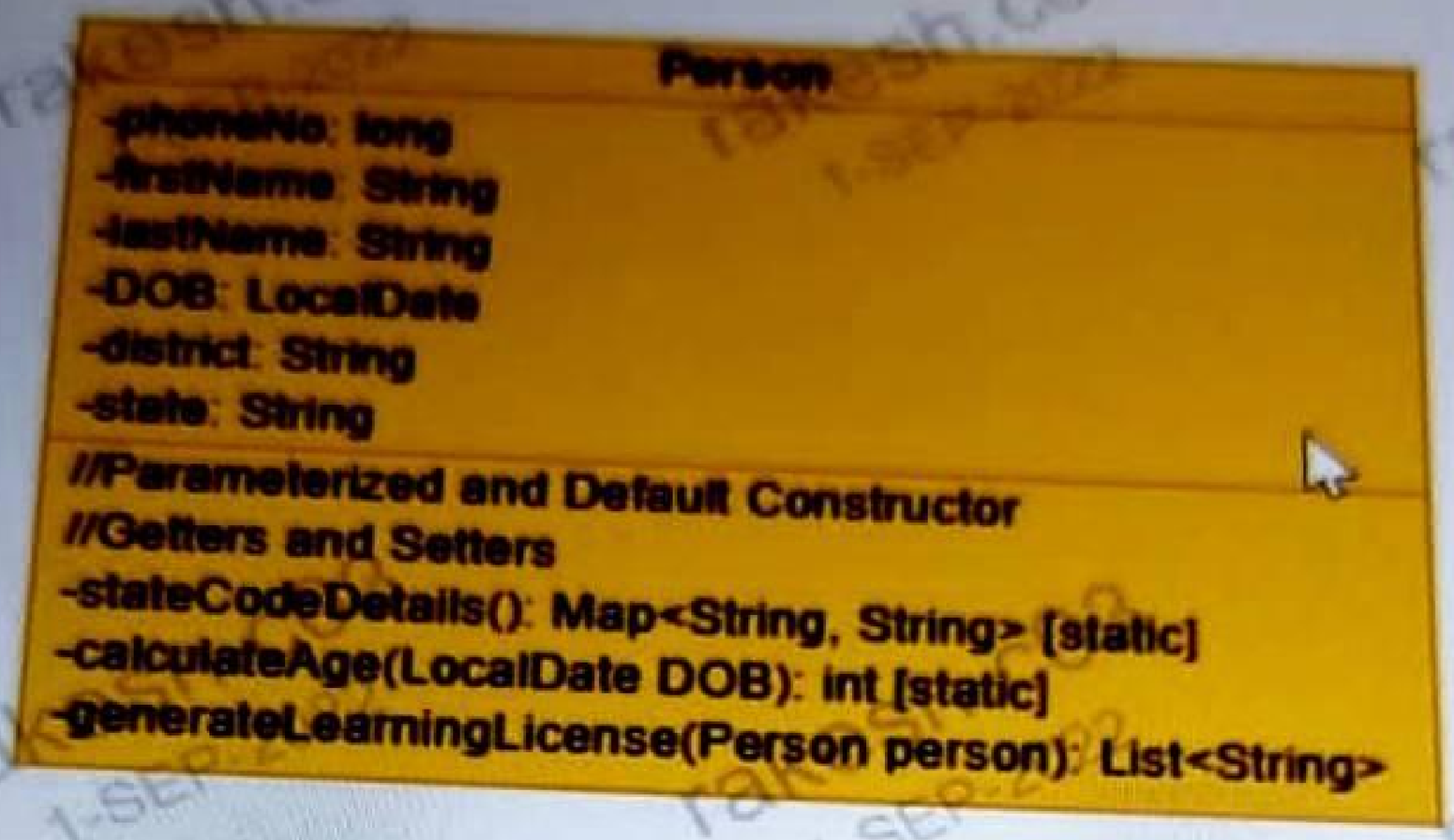
Requirement:

The automation process should return the full name, age, learner license number, expiry date of Learner License, and a success message.

Class Diagram:



Class Diagram:



Instructions:

As per the above requirement, the **Main** class accepting all the above parameters with the proper function calls is already provided to you.

A partially implemented **Person** class is given. Complete the **generateLearnerLicense(Person person)** in the **Person** class to complete the automation process.

Main.java(Fully implemented):

- Logic is already provided to take the **person** details like- '**phoneNo**', '**firstName**', '**lastName**', '**DOB**', '**state**', '**district**' from the keyboard, and with help of these values a '**person**' object is created by using the parameterized constructor of **Person**

complete the automation process.

Main.java(Fully implemented):

- Logic is already provided to take the **person** details like- **'phoneNo'**, **'firstName'**, **'lastName'**, **'DOB'**, **'state'**, **'district'** from the keyboard, and with help of these values a **'person'** object is created by using the parameterized constructor of **Person** class.
- Next, there is a method call to **generateLearnerLicense(Person person)** which is returning a **List** of **String(s)**. And we are just printing the returned **list** as output.

Person.java(Partially implemented):

- This class has instance variables like - **'phoneNo'**, **'firstName'**, **'lastName'**, **'DOB'**, **'state'**, **'district'**.
- This class contains both default and parameterized constructors.
- The **toString**() function from the Object class is overridden here.
- It has various **private**, **public**, and **static** methods which are explained below.

private static Map<String, String> stateCodeDetails()[Fully implemented]:

- This method is used to return the **Map<String, String>** of **states** and **respective state codes**.
- This method is **private** and **static**.

private static Map<String, String> stateCodeDetails()[Fully implemented]:

- This method is used to return the **Map<String, String>** of **states** and **respective state codes**.
- This method is **private** and **static**.
- This method will return the state codes for **six states** for which the automation has completed.
- The states are- **'Karnataka'** with state code **'KA'**, **'Kerla'** with state code **'KL'**, **'Maharastra'** with state code **'MH'**, **'Tamilnadu'** with state code **'TN'**, **'Andhrapradesh'** with state code **'AP'**, and **'Odisha'** with state code **'OD'**.

public List<String> generateLearnerLicense(Person person) [Patially implemented]:

- Step:1
  - Create an **empty list** of **String** named **'learnerLicenceDetails'** which will hold the details of **Learner License**.
  - Calculate the **age** by using the **private static calculateAge(LocalDate DOB)** method in the **Person** class by using the '**person'** object which is passed as the **first parameter**.
- Step:2
  - If the age of the **person** is **less than18** then the set an error message as 'Learner License can't be generated as

lic List<String> generateLearnerLicense(Person person) [Patially lemented]:

- Step:1
  - Create an **empty list** of **String** named '**learnerLicenceDetalls**' which will hold the details of **Learner License**.
  - Calculate the **age** by using the **private static calculateAge(LocalDate DOB)** method in the **Person** class by using the '**person**' object which is passed as the **first parameter**.
- Step:2
  - If the age of the **person** is **less than18** then the set an **error message** as '**Learner License can't be generated as age is less than 18**'.
  - **Add** the same **error message** in the '**learnerLicenceDetails**' string variable.
  - **Return** the '**learnerLicenceDetails**'.
- Step:3
  - Step:3.1: Else create a String variable as '**fullName**' and populate the same variable with the **firstName** and the **lastName** of the **person** separated by a **space**.
    - Example- If the **firstName** of the **person** is '**Sachin**' and the **lastName** of the **person** is '**Tendulkar**', then the '**fullName**' will be populated as '**Sachin Tendulkar**'.
  - Step:3.2:**Add** the same '**fullName**' to the 'learnerLicenceDetails'.
  - Step:3.3:Convert the **Person's age** calculated in **Step1** into

- **Step3.4.1**: Get all the **state code details** by calling the previously implemented **private static** method **'stateCodeDetails()'** and store it in a new **Map<String, String>**.
- **Step3.4.2**:Create a new **String** variable **'stateCodeOfThePerson'** which will store the **2-character state code** of the **Person** and initialize it to **null**.
- **Step3.4.3**:**Traverse** the same **Map<String, String>** obtained in **Step3.4.1** containing the **state code** details and check if the **person's state matches** with any of the **key** of the **Map<String, String>**. If matches, then store the **2-character state code** value in the variable '**stateCodeOfThePerson**'.
- **Step3.4.4**:**Check** if '**stateCodeOfThePerson**' is **null**. If it's **null**, then the automation process for the given **state** is **incomplete**. So, **re-initialize** the '**learnerLicenceDetails**'. Add an **error message** as "**Your state not found in our DB, So the LL can't be generated**". And, return the '**learnerLicenceDetails**'.
- **Step3.4.5:** Create a **String** variable as '**districtCode**' and populate it with the **first two characters** of the **person's district** in **uppercase**.
- Step3.4.6: Generate the **Learner License Number** as stateCode+districtCode+ the phone number of the **person.** And add it to '**learnerLicenceDetails**'.

- Step 4
  - Set the validity of the **Learner License** as **40 years** from

- Step 4
  - Set the validity of the **Learner License** as **40 years** from the **person's DOB.**
  - **Convert** the same into **String** format in **'yyyy-mm-dd'** format and **add** it to **'learnerLicenceDetails'**
  - **Note:** Only calling a **toString()** function on a **LocalDate** variable will be get converted into **'yyyy-mm-dd'** format.
- Step 5
  - Add a success message as **"Your driving test will be scheduled within next 7 days."**
  - Add the same success message to the **'learnerLicenceDetails'**.
- Step 6
  - Finally, **return** the **'learnerLicenceDetails'**.

Partially implemented Person.java, and fully implemented Main.java, are already given to you.

Note:

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

Languages: Java

Sample Input	Sample Output	Explanation
7022713766 John Wick 13/06/2000 Karnataka Mysuru	[John Wick, 21, KA-MY-7022713766, 2040-06-13, Your driving test will be scheduled within next 7 days.]	Output came as per the instructions.
7022713766 John Wick 13/06/2021 Karnataka Mysuru	[Learner License can't be generated as age is less than 18]	Output came as per the instructions.
7022713766 John Wick 13/06/2000 Punjab Mysuru	[Your state not found in our DB, So the LL can't be generated]	Output came as per the instructions.

Languages: Java

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

```java
package skillpac2;

import java.time.LocalDate;

class Persons {
    private long phoneNo;
    private String firstName;
    private String lastName;
    private LocalDate DOB;
    private String state;
    private String district;

    public Persons() {

    }

    public Persons(long phoneNo, String firstName, String lastName, LocalDate dOB, String state, String district) {
        super();
        this.phoneNo = phoneNo;
        this.firstName = firstName;
        this.lastName = lastName;
        DOB = dOB;
        this.state = state;
        this.district = district;
    }

    public long getPhoneNo() {
        return phoneNo;
    }

    public void setPhoneNo(long phoneNo) {
```

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

```
    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public LocalDate getDOB() {
        return DOB;
    }

    public void setDOB(LocalDate dOB) {
        DOB = dOB;
    }

    public String getState() {
        return state;
    }

    public void setState(String state) {
        this.state = state;
    }

    public String getDistrict() {
        return district;
    }

    public void setDistrict(String district) {
        this.district = district;
    }
```

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

```
79
80      public void setDistrict(String district) {
81          this.district = district;
82      }
83
84      @Override
85      public String toString() {
86          return "Persons [phoneNo=" + phoneNo + ", firstName=" + firstName + ", lastName=" + lastName + ", DOB=" + DOB
87                  + ", state=" + state + ", district=" + district + "]";
88      }
89
90      private static Map<String, String> stateCodeDetails() {
91          Map<String, String> stateCodeMap = new HashMap<>();
92          stateCodeMap.put("Karnataka", "KA");
93          stateCodeMap.put("Kerala", "KL");
94          stateCodeMap.put("Maharastra", "MH");
95          stateCodeMap.put("TamilNadu", "TN");
96          stateCodeMap.put("AndraPradesh", "AP");
97          stateCodeMap.put("Odhisha", "OD");
98          return stateCodeMap;
99
100     }
101
102     private static int calculateAge(LocalDate DOB) {
103         return Period.between(DOB, LocalDate.now()).getYears();
104
105     }
106
107     public List<String> generateLearnerLicense(Persons person) {
108         List<String> learnerLicenseDetails = new ArrayList<String>();
109         int age = calculateAge(person.getDOB()) - 1;
```

```
106
107     public List<String> generateLearnerLicense(Persons person) {
108         List<String> learnerLicenseDetails = new ArrayList<String>();
109         int age = calculateAge(person.getDOB()) - 1;
110         if (age < 18) {
111             learnerLicenseDetails.add("Learner License can't be generated as age is less than 18");
112             return learnerLicenseDetails;
113         }
114         String fullName = person.getFirstName() + " " + person.getLastName();
115         learnerLicenseDetails.add(fullName);
116         Integer age1 = age;
117         learnerLicenseDetails.add(age1.toString());
118
119         Map<String, String> stateCode = stateCodeDetails();
120         String stateCodeOfThePerson = null;
121         if (stateCode.keySet().contains(person.getState())) {
122             for (Map.Entry<String, String> map : stateCode.entrySet()) {
123                 if (map.getKey().equals(person.getState())) {
124                     stateCodeOfThePerson = map.getValue();
125                 }
126             }
127         }
128         if (stateCodeOfThePerson == null) {
129             learnerLicenseDetails = new ArrayList<String>();
130             learnerLicenseDetails.add("your State not found in our DB,so the LL can't be generated");
131             return learnerLicenseDetails;
132         }
133         String districtCode = person.getDistrict().substring(0, 2).toUpperCase();
134         String learnerLicenseNumber = stateCodeOfThePerson + "-" + districtCode + "-" + person.getPhoneNo();
135         learnerLicenseDetails.add(learnerLicenseNumber);
136         String valid = person.getDOB().plusYears(40).toString();
```

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

```
        String districtCode = person.getDistrict().substring(0, 2).toUpperCase();
        String learnerLicenseNumber = stateCodeOfThePerson + "-" + districtCode + "-" + person.getPhoneNo();
        learnerLicenseDetails.add(learnerLicenseNumber);
        String valid = person.getDOB().plusYears(40).toString();
        learnerLicenseDetails.add(valid);
        String success = "Your driving test will be scheduled within next 7 days";
        learnerLicenseDetails.add(success);

        return learnerLicenseDetails;
    }
}

public class Employee {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        long phoneNo = input.nextLong();
        String fName = input.next();
        String lName = input.next();
        String DOB = input.next();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
        LocalDate dob = LocalDate.parse(DOB, formatter);
        String state = input.next();
        String district = input.next();
        Persons person = new Persons(phoneNo, fName, lName, dob, state, district);
        List<String> learnerLicenseDetails = new Persons().generateLearnerLicense(person);
        System.out.println(learnerLicenseDetails);
        input.close();
    }
}

```

```
141             return learnerLicenseDetails;
142         }
143 }
```

Javadoc | Declaration | Console | JUnit

<terminated> Employee (1) [Java Application] D:\TypeScript SDK 4.5.5\SoftwareCenterApplications\Eclipse IDE for Java Developers\eclipse-java

```
7022713766
John
Wick
13/06/2000
Karnataka
Mysuru
[John Wick, 21, KA-MY-7022713766, 2040-06-13, Your driving test will be scheduled within next 7 days]
```