**КОНКУРСНОЕ**

**ЗАДАНИЕ**

**Сессия 1**

# Описание задачи

## Разработка библиотеки классов

Вашей задачей является разработка двух библиотек, которые позволят Вам использовать их в следующих сессиях при выполнении конкурсного задания. При создании библиотек Вы должны использовать только те именования, которые указаны в задании. Если названия или типы данных будут отличны от указанных, Ваша работа не сможет быть проверенной.

Обращаем Ваше внимание, проверка библиотек автоматическая!

Во время выполнения конкурсного задания вам необходимо использовать предоставленные ресурсы.

Библиотеки не должны работать с базой данных. Все данные должны храниться только в рамках библиотеки.

### Регистрационный знак транспортных средств

Вашей задачей является создание библиотеки REG\_MARK\_LIB.dll или REG\_MARK\_LIB.jar, которая имеет следующие методы:

1. **Boolean CheckMark(String mark)** – данный метод проверяет переданный номерной знак в формате a999aa999 (латинскими буквами) и возвращает true или false в зависимости от правильности номерного знака. Метод должен учитывать также и существующие номера регионов.
2. **String GetNextMarkAfter(String mark)** – данный метод принимает номерной знак в формате a999aa999 (латинскими буквами) и выдает следующий номер в данной серии или создает следующую серию.
3. **String GetNextMarkAfterInRange(String prevMark, String rangeStart, String rangeEnd)** – данный метод принимает номерной знак в формате a999aa999 (латинскими буквами) и выдает следующий номер в данной данном промежутке номеров rangeStart до rangeEnd (включая обе границы). Если нет возможности выдать следующий номер, необходимо вернуть сообщение “out of stock”.
4. **Integer GetCombinationsCountInRange(String mark1, String mark2)** – данный метод принимает два номера в формате a999aa999 (латинскими буквами) и возвращает количество возможных номеров между ними (включая обе границы). Метод необходим, чтобы рассчитать оставшиеся свободные номера для региона.

Полную информацию о регистрационном знаке Вы найдете в файле vehicle\_registration\_plates.

## Разработка модульных тестов (Unit-tests)

Для выполнения процедуры тестирования созданного вами метода библиотеки, вам необходимо создать отдельный проект модульных тестов. В рамках проекта разработайте тесты, максимально полно покрывающие функционал метода. Ничего страшного, если ваш метод работает не совсем идеально и тесты могут быть не пройдены в связи с этим - в данном модуле это не так важно. Обратите внимание, что имена тестов должны отражать их суть. Необходимо разработать модульные тесты, которые на основании исходных данных можно условно разделить на 2 группы следующим образом: 10 методов низкой сложности и 5 методов высокой сложности.

## Разработка тестовых сценариев (Test-cases)

Для выполнения процедуры тестирования редактирования продукции Вам нужно описать пять сценариев. Удаление может быть выполнимо, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон testing-template.docx.