**КОНКУРСНОЕ**

**ЗАДАНИЕ**

**Сессия 1**

# Описание задачи

1. Разработка библиотек

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.

### VIN

Ваша задачей является создание библиотеки VIN\_LIB.dll или VIN\_LIB.jar , которая имеет следующие методы:

1. **Boolean CheckVIN(String vin)** – данный метод проверяет переданный VIN номер и возвращает true или false в зависимости от правильности VIN номера. VIN номер должен удовлетворять всем требованиям, а также у него должна сходиться контрольная сумма.
2. **String GetVINCountry(String vin)** – данный метод возвращает страну, в которой было изготовлено транспортное средство.
3. **Integer GetTransportYear(String vin)** – данный метод возвращает год, в который было выпущено транспортное средство.

Полную информацию о VIN Вы найдете в файле vin\_info.

### 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.

## Загрузка данных

На данном этапе вы должны так же начать проектировать и разрабатывать базу данных. В ресурсах к сессии Вам предоставлены данные, которые необходимо загрузить в вашу базу данных. Они необходимы при дальнейшей разработке системы. Ознакомьтесь с предоставленными данными в архиве import\_data и загрузите их в базу данных, создав необходимые сущности.

## Авторизация

В связи с тем, что система хранит и обрабатывает персональные данные людей, занесенных в базу, то необходимо предусмотреть систему их защиты. Минимальным шагом для этого является система авторизации.   
Так как система только разрабатывается, то для авторизации мы будем использовать только одного пользователя – inspector. Пароль должен совпадать с логином.

Для защиты от подборов паролей, необходимо ограничить количество попыток ввода с неправильным паролем. Если пользователь вводит неправильные данные более чем 3 раза, необходимо заблокировать ввод данных на 1 минуту. Блокировка остается даже если пользователь сразу же перезапускает приложение.

Если приложением не пользуются более одной минуты, то необходимо закрыть все открытые окна приложения и вернуть пользователя на окно авторизации.