**Тестовое задание “Справочник автомобилей”**

Ссылка на github <https://github.com/hramovalexey/Car-directory>

1. Общая информация

Задание выполнено в виде консольного приложения на Java.

Используется сторонний парсер для разбора Json запроса JSONParser (см. п.3)

Хранилище представляет собой базу данных «AutoDB» в PostgreSQL c единственной таблицей в ней «auto\_list». Предполагается, что база данных и таблица уже существуют.

SQL запрос для создания нужной таблицы см. в файле reset\_table.sql

Запросы для заполнения тестовой таблицы см. в файле populate\_table.sql

В приложении производится подключение к моей локальной базе данных. Выбор реквизитов для подключения к другой базе во время работы никак не реализован и может быть произведен корректировкой реквизитов в конструкторе класса AutoList.

Лог пишется в локальный файл carLog.txt. В архиве пример файла приведен.

Результаты транзакций («Успешно», «Объект не найден» и пр.) выводятся в консоль (в задании не было уточнено, что с ними делать).

Статистика выводится в минимальном объеме согласно заданию.

Из дополнительных заданий реализовано:

- лог

- статистика

1. Описание API

Доступ к API осуществляется путем вызова методов экземпляров класса AutoList:

- Конструктор создает соединение с базой данных, запускает лог. Если локальный файл carLog.txt существует, то продолжает писать в него, если - нет, то создает новый

**public AutoList()**

* добавление авто.

str - Json запрос вида:

{ "number" : "A020BE96", "model" : "Wolksvagen Tiguan", "color" : "Зеленый", "year" : 2016}

**public void addCar(String str)**

- удаление авто. str - номер автомобиля

**public void deleteCar(String str)**

- текущий список авто с параметрами

str - Json запрос вида

{ "model" : "Datsun", "color" : "Белый", "year" : 2016}

в примере будут выведены все белые Datsun 2016го года и отсортированы в порядке model, color, year

**public void listCar(String str)**

- Текущий список авто без параметров

**public void listCar()**

- Показать статистику:

- Количество записей

- Дата первой записи

- Дата последней записи

**public void getStats()**

- Вывести лог

**public void printLog()**

1. Краткое описание структуры

Доступ к API осуществляется путем вызова методов экземпляров класса AutoList.java (см. п.2)

Для работы приложения написаны вспомогательные классы:

- Log.java - Вспомогательный класс для ведения лога в локальном файле

Если файл carLog.txt существует, то продолжает писать в него,

если - нет, то создает новый

- Auto.java - Вспомогательный класс для объекта Auto (методы по манипуляциям с характеристиками авто)

- ResultCarList.java - Класс для формирования списка результатов

Кроме того, используется сторонний парсер для разбора Json запроса JSONParser (гитхаб <https://github.com/fangyidong/json-simple>)