**ЗАДАНИЕ 2**

1. Массивы (Работа с парком машин)

• Создайте массив, представляющий годы выпуска 50 случайных машин (от 2000 до 2025).  
 • Выведите только машины, выпущенные после 2015 года.  
 • Посчитайте средний возраст авто.

2. Коллекции (Управление моделями)

• Создайте список с названиями моделей машин (например: *Toyota Camry, BMW X5*). Могут быть дубликаты!  
 • Удалите дубликаты, затем отсортируйте модели в обратном алфавитном порядке, выведите на экран, затем сохраните в Set.  
 • Реализуйте проверку: если модель содержит слово "Tesla", замените её на "ELECTRO\_CAR".

3. equals/hashCode (Сравнение автомобилей)

• Создайте класс Car с полями: VIN, модель, производитель, год выпуска, пробег, цена

• Переопределите equals и hashCode, чтобы две машины считались одинаковыми только при совпадении VIN.  
 • Добавьте в HashSet несколько машин (включая дубликаты по VIN) и убедитесь, что дубликаты не добавляются.  
 • Реализуйте Comparable<Car> для сортировки по году выпуска (от новых к старым).\*

4. Stream API (Анализ автопарка)

Дан список машин (List<Car>):  
 • Отфильтруйте только машины с пробегом меньше 50\_000 км (добавьте поле mileage).  
 • Отсортируйте по цене (по убыванию).  
 • Выведите топ-3 самые дорогие машины.  
 • Посчитайте средний пробег всех машин.  
 • Сгруппируйте машины по производителю в Map<String, List<Car>>.

5. Практическое задание: Автоцентр (Реализация системы) \*

Реализуйте класс CarDealership, содержащий список автомобилей. У каждой машины:

* VIN, модель, производитель, год выпуска, пробег, цена, тип (enum: SEDAN, SUV, ELECTRIC и т.д.).

Методы:

1. Добавить машину в автоцентр (проверять дубликаты по VIN).

2. Найти все машины указанного производителя (использовать Stream).

3. Вывести среднюю цену машин определённого типа (SUV, ELECTRIC и др.).

4. Вернуть список машин, отсортированных по году выпуска (от новых к старым).

5. Дополнительно: реализовать статистику:

o Количество машин каждого типа.

o Самая старая и самая новая машина в наличии.

Создайте меню с возможностью вызова каждого метода (например, через Scanner).

**Ссылки на доп. Изучение:**

1. <https://metanit.com/java/tutorial/10.1.php> - Stream API

2. <https://metanit.com/java/tutorial/5.1.php> - По работе с коллекциями