

## Домашнее задание по теме «Java Collections. Stream API»

### Формулировка задания:

1. Реализовать класс Автомобиль. У класса есть поля, свойства и методы.

Поля класса:

- а) Номер автомобиля;
- б) Модель;
- в) Цвет;
- г) Пробег;
- д) Стоимость.

Обратить внимание на переопределение метода toString, на сеттеры и геттеры, модификаторы доступа полей.

2. Проверить работу в классе Main, методе main.

3. Создать объект Java Collections со списком автомобилей.

4. Используя Java Stream API, вывести (можно сделать любые 2 пункта из 4):

1) Номера всех автомобилей, имеющих заданный в переменной цвет colorToFind или нулевой пробег mileageToFind.

- 2) Количество уникальных моделей в ценовом диапазоне от  $n$  до  $m$  тыс.
- 3) Вывести цвет автомобиля с минимальной стоимостью.
- 4) Среднюю стоимость искомой модели `modelToFind`

Car.java		App.java		input.txt	
1	[НОМЕР_АВТОМОБИЛЯ][МОДЕЛЬ][ЦВЕТ][ПРОБЕГ][ЦЕНА]				
2	a123me Mercedes White 0 8300000				
3	b873of Volga Black 0 673000				
4	w487mn Lexus Gray 76000 900000				
5	p987hj Volga Red 610 704340				
6	c987ss Toyota White 254000 761000				
7	0983op Toyota Black 698000 740000				
8	p146op BMW White 271000 850000				
9	u893ii Toyota Purple 210900 440000				
10	l097df Toyota Black 108000 780000				
11	y876wd Toyota Black 160000 1000000				

```
1 package homeworks.homework11;
2
3 import java.util.Objects;
4
5 public class Car { ② programakc
6     private String number; 7 usages
7     private String model; 7 usages
8     private String color; 7 usages
9     private Integer mileage; 7 usages
10    private Integer cost; 7 usages
11
12    public Car() { ② programakc
13    }
14
15    public Car(String number, String model, String color, Integer mileage, ② programakc
16        Integer cost) {
17        this.number = number;
18        this.model = model;
19        this.color = color;
20        this.mileage = mileage;
21        this.cost = cost;
22    }
23
24    public String getNumber() { no usages ② programakc
25        return number;
26    }
27
```

```
28 public void setNumber(String number) { no usages 2 programakc
29     this.number = number;
30 }
31
32 public String getModel() { 1 usage 2 programakc
33     return model;
34 }
35
36 public void setModel(String model) { no usages 2 programakc
37     this.model = model;
38 }
39
40 public String getColor() { 1 usage 2 programakc
41     return color;
42 }
43
44 public void setColor(String color) { no usages 2 programakc
45     this.color = color;
46 }
47
48 public Integer getMileage() { no usages 2 programakc
49     return mileage;
50 }
51
52 public void setMileage(Integer mileage) { no usages 2 programakc
53     this.mileage = mileage;
54 }
```

```
55
56 public Integer getCost() { 2 usages  ⓘ programakc
57     return cost;
58 }
59
60 public void setCost(Integer cost) { no usages  ⓘ programakc
61     this.cost = cost;
62 }
63
64 @Override ⓘ programakc
65 public boolean equals(Object o) {
66     if (o == null || getClass() != o.getClass()) return false;
67     Car car = (Car) o;
68     return number == car.number && Objects.equals(model, car.model) &&
69         Objects.equals(color, car.color) && Objects.equals(mileage,
70             car.mileage) && Objects.equals(cost, car.cost);
71 }
72
73 @Override ⓘ programakc
74 public int hashCode() {
75     return Objects.hash(number, model, color, mileage, cost);
76 }
77
78 @Override ⓘ programakc
79 public String toString() {
80     return number + " " + model + " " + color + " " + mileage + " " + cost;
81 }
82 }
```

```

1 package homeworks.homework11;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6 import java.util.ArrayList;
7 import java.util.Comparator;
8 import java.util.List;
9
10 public class App { new *
11     public static void main(String[] args) { new *
12         List<Car> cars = new ArrayList<>();
13         String title = "";
14
15         try (BufferedReader bufferedReader = new BufferedReader(new FileReader(
16             "src/homeworks/homework11/input.txt"))) {
17
18             title = bufferedReader.readLine();
19             String[] readData;
20             String str = "";
21
22             while (str != null) {
23                 str = bufferedReader.readLine();
24                 if (str != null) {
25                     readData = str.split("\\|");
26
27                     cars.add(new Car(readData[0], readData[1], readData[2],
28                                     Integer.parseInt(readData[3]), Integer.parseInt(readData[4])));
29                 }

```

```
30     }
31
32     } catch (IOException e) {
33         throw new RuntimeException(e);
34     }
35
36     System.out.println("\nАвтомобили в базе:\n" + title);
37
38     for (Car c : cars) {
39         System.out.println(c.toString());
40     }
41
42     String getCarByColorWithMinCost = cars.stream() Stream<Car>
43         .min(Comparator.comparingInt(Car::getCost)) Optional<Car>
44         .map(Car::getColor) Optional<String>
45         .get();
46
47     System.out.println("\nЦвет автомобиля с минимальной стоимостью: "
48         + getCarByColorWithMinCost);
49
50     String modelToFind = "Toyota";
51
52     Double getAverageCostByCarModel = cars.stream() Stream<Car>
53         .filter( Car e -> e.getModel().equals(modelToFind))
54         .mapToDouble(Car::getCost) DoubleStream
55         .average() OptionalDouble
56         .getAsDouble();
```



```
57
58     System.out.println("\nСредняя стоимость модели " + modelToFind + " "
59         + getAverageCostByCarModel);
60 }
61 }
```

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" -javaagent:G:\JetBrains\IntelliJ\Idea2025.1\lib\idea_rt
.jar=51815 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath
D:\JavaProject\JavaDevHomeworks\out\production\JavaDevHomeworks homeworks.homework11.App
```

Автомобили в базе:

[НОМЕР_АВТОМОБИЛЯ]	[МОДЕЛЬ]	[ЦВЕТ]	[ПРОБЕГ]	[ЦЕНА]
a123me	Mercedes	White	0	8300000
b873of	Volga	Black	0	673000
w487mn	Lexus	Gray	76000	900000
p987hj	Volga	Red	610	704340
c987ss	Toyota	White	254000	761000
0983op	Toyota	Black	698000	740000
p146op	BMW	White	271000	850000
u893ii	Toyota	Purple	210900	440000
l097df	Toyota	Black	108000	780000
y876wd	Toyota	Black	160000	1000000

Цвет автомобиля с минимальной стоимостью: Purple

Средняя стоимость модели Toyota 744200.0