# 03. Automotive Repair Shop



*You are a manager of an automotive repair shop, and you need a monitoring program for the vehicles.*

## Preparation

Download the skeleton provided in Judge. **Do not** change the **packages**!

**Pay attention to name the package automotiveRepairShop, all the classes, their fields, and methods the same way they are presented in the following document. It is also important to keep the project structure as described.**

## Problem Description

Your task is to create a repository that stores vehicles by creating the classes described below.

### Vehicle

You are given a class **Vehicle** with the following properties:

* **VIN: String**
* **mileage: int**
* **damage: String**

The class **constructor** should receive **VIN, mileage** and **damage**.

Override the **ToString()** method in the following format:  
**"Damage: {damage}, Vehicle: {VIN} ({mileage} km)"**

### RepairShop

**Next**, you are given a class **RepairShop** that has **vehicles** (a List that stores vehicles). All entities inside the repository have the **same properties**. The **RepairShop** class should have the following **properties**:

* **capacity: int**
* **vehicles: List<Vehicle>**

The class **constructor** should receive **capacity**, also it should initialize the **Vehicles** with a new instance of the collection.Implement the following features:

* **Method addVehicle(Vehicle vehicle)** – **adds** an **entity** to the **collection** of Vehicles, **if** the **Capacity** **allows it**.
* **Method removeVehicle(String vin)** – **removes** a **vehicle by** **given vin,** if such **exists**, and **returns boolean** (**true** if it is removed, otherwise – **false**)
* **Method getCount()** – **returns** the number of **vehicles**, **registered** in the RepairShop
* **Method getLowestMileage()** – **returns** the **Vehicle** with the **lowest value of Mileage property.**
* **Method report()** – **returns** a **string** in the following **format**:
  + **"Vehicles in the preparatory:  
    {Vehicle1}  
    {Vehicle2}  
    (…)"**

### Constraints

* The **VIN** of the vehicles will be **always unique**.
* You will always have vehicles added before receiving methods manipulating the vehicles in the RepairShop.

### Examples

This is an example of how the **RepairShop** class is **intended to be used**.

|  |
| --- |
| **Sample code usage** |
| //Initialize the repository (RepairShop)RepairShop repairShop = **new** RepairShop(5);  //Initialize entity (Vehicle)Vehicle vehicle1 = **new** Vehicle(**"**1HGCM82633A123456**"**, 50000, **"**Oil leakage**"**);  //Print VehicleSystem.out.println(vehicle1); // Damage: Oil leakage, Vehicle: 1HGCM82633A123456 (50000 km)//Add VehiclerepairShop.addVehicle(vehicle1);  //Remove VehicleSystem.out.println(repairShop.removeVehicle(**"**1HGCM82633A123459**"**)); // false  System.out.println(repairShop.removeVehicle(**"**1HGCM82633A123456**"**)); // trueVehicle vehicle2 = **new** Vehicle(**"**5YJSA1CN7DFP12345**"**, 80000, **"**Overheating issue**"**);  Vehicle vehicle3 = **new** Vehicle(**"**JM1GJ1W56F1234567**"**, 120000, **"**Coolant leakage**"**);  Vehicle vehicle4 = **new** Vehicle(**"**2C3CDXAT4CH123456**"**, 95000, **"**Timing belt failure**"**);  Vehicle vehicle5 = **new** Vehicle(**"**WAUZZZ8K9FA123456**"**, 66000, **"**Cylinder misfire**"**);  Vehicle vehicle6 = **new** Vehicle(**"**1G1BL52P3RR123456**"**, 150000, **"**Transmission failure**"**);  Vehicle vehicle7 = **new** Vehicle(**"**JTDKB20U993123456**"**, 65000, **"**Piston damage**"**);  //Add More VehiclesrepairShop.addVehicle(vehicle2);  repairShop.addVehicle(vehicle3);  repairShop.addVehicle(vehicle4);  repairShop.addVehicle(vehicle5);  //Get CountSystem.out.println(repairShop.getCount()); //4  repairShop.addVehicle(vehicle6);  repairShop.addVehicle(vehicle7);  //Get CountSystem.out.println(repairShop.getCount()); //5  //Get LowestMileageSystem.out.println(repairShop.getLowestMileage()); //Damage: Cylinder misfire, Vehicle: WAUZZZ8K9FA123456 (66000 km)  //Report System.out.println(repairShop.report());  //Vehicles in the preparatory:  //Damage: Overheating issue, Vehicle: 5YJSA1CN7DFP12345 (80000 km)  //Damage: Coolant leakage, Vehicle: JM1GJ1W56F1234567 (120000 km)  //Damage: Timing belt failure, Vehicle: 2C3CDXAT4CH123456 (95000 km)  //Damage: Cylinder misfire, Vehicle: WAUZZZ8K9FA123456 (66000 km)  //Damage: Transmission failure, Vehicle: 1G1BL52P3RR123456 (150000 km) |

## Submission

Zip all the files in the project folder except the **bin** and **obj** folders

Submit a **single .zip file**, containing **automotiveRepairShop package, with the classes inside (RepairShop, Vehicle, and the Main class)**, there is no specific content required inside the Main class e. g. you can do any kind of local testing of your program there. However, there should be a **main(String[] args)** method inside.