

Henry Ford assembly line

Source code must be submitted in gitlab.talentboost.org server. *Your task will be assessed in the following categories: Correctness (validation of input parameters, proper exception handling), Coding Style (java docs, formatting, readability and naming conventions), Design (Interfaces, separations of concerns, proper model and patterns), Algorithms and Unit tests.*

Your task is to design and implement a Vehicle Assembly line simulator. The main purpose of the assembly line is to manufacture cars. Your software will allow the customer to track all cars manufactured by the line. Sales and marketing departments will issue requests to your software when they need to track an action done with a car. You should be able to support the following: assemble (create) cars, disassemble (remove) cars and print (display) information about manufactured and disassembled cars.

A famous Car manufacturer is starting one instance of your assembly line in one of their facilities in Bulgaria. They offer a variety of models and engines to their customers and they hope to become more efficient by using your software.

Each car that is coming out from your assembly line should have a unique VIN (Vehicle Identification Number), Model, Engine, and Transmission.

The **VIN** is an alphanumeric string of 17 symbols, excluding the letters I, O and Q (to avoid confusion with the numbers 1 and 0). For the purpose of this task consider the

first two symbols to be the ISO code for the country where the car is produced (e.g. BG for Bulgaria)[They permit excluded symbols and must be capital], the third symbol to be a unique factory number for the country (one of {0-9}). The other 14 symbols are factory specific unique identifier of each car.

Our car manufacturer offers the following **models**: A1, A2, A3, A4, A5, A6, A7 and A8. All these can be either sedan(default type), hatchback or kombi.

Each model can be equipped with Petrol or Diesel **engine**. Each engine has a specific displacement measured in cubic centimeters, which correlates with the engine's power (measured in kW) as given on the following table of equivalent

Engine displacement (cc)	Power in kW
1000	74
2000	134
3000	245

4000	253
5000	331
6000	510
8000	736

Note: Diesel engines cc limitation: $2000 \leq cc \leq 6000$

If a Turbo is mounted to any of the engines above, the power increases with 30%. Engines should comply to one of the following emission standards: euro3, euro4, euro5, euro6. .

Any of the following **transmissions** can be fitted into a car:

Automatic	Manual
4 - gears	4 – gears
5 - gears	5 – gears
6 - gears	6 – gears
8 - gears	n/a

Your assembly line should accept the following commands from keyboard:

- `create car <specification>` - creates a car based on the specification and adds it to persistent storage

Example input and output:

Input:

```
create car model=A5{-hatchback} engine=P|D|E{-100hp|6L{-T} {-euro3}}
{transmission=Auto|Manual{-4}}
```

Output:

```
vin          |model |type      |engine          |transmission|status
BG212345678901234 |A5     |hatchback |D-130hp{T}-euro3|Auto-4      |as/disas
```

- `disassemble <vin>` - deletes/removes a vehicle from database with the provided vin
- `print {vin|all}` – displays information about a specific vehicle for given vin or a list of all manufactured vehicles in the following format

```
vin          |model |type      |engine          |transmission |status
BG212345678901234 |A5     |hatchback |D-100hp{T}-euro3 |Auto-4
|as/disas
```

****notes****

If no parts correspond to the specified, ..?

VIN cannot be used twice

All assembled cars must be recorded

All arguments between curly brackets are **optional**!

D-179hp-T:

1. Convert to kW

179hp -> 133.48 kW

2. Check in table if there is an engine with this power - tolerance: +/-1 kW

3. If there is a value close or exact as wanted engine power we create the engine else throw error.

4.If turbo is wanted - add 30% to current power

30% from engine power -> 44.74 kW(nearly)

133.48 + 44.74 = 178,22 kW total engine power that we print in the end.

Extension 1:

1.1) Our car manufacturer is very happy with the production line, but they want your program to read the commands from a file. They aggregate all purchases from different sales offices in a single file and they want to give this file to the assembly line. Vehicles on the assembly line should be created one by one - in the same order as specified in the file.

1.2) The second requirement is that the assembly line must be able to manufacture their SUV models. Basically you need to extend your software to support vehicles of type suv. Suv models are Q1, Q2, Q3... Q8 and they are using the same engines and transmissions as cars.

Example input and output:

Input:

```
create car|suv model=A5|Q1{-hatchback} engine=P|D{-100hp|6L{-T}-euro3}
{transmission=Auto{-4}}
```

Output:

```
vin          |model |type      |engine          |transmission
BG212345678901235 |Q1     |hatchback |D-130hp{T}-euro3 |Auto-4
```

//comment vehicle type (car, suv) should be stored, output - optional

Extension 2:

The company discovered that all vehicles with 2L Euro4 diesel engines do not comply to the standard due to bug in engine ECU. Because of this they need to recall all vehicles and to update the ECU firmware, which will make these vehicles – Euro5 compliant.

You need to be able to find vehicle/s by specification. The customer has requested a “find” command, which finds all cars according to given emission standard and prints their info.

In order to update the vehicles, the customer has also requested an “update” command, which updates the emission standard of a a car specified by its vin.

Input:

Find euro4

update <vin> engine=*-*euro5 # updates a vehicle with given to euro5 standard.

update <vin> engine=*-*T-* #adds a turbo to a vehicle with give vin

update <vin> engine=*-*T-* transmission=Auto{-4} #adds a turbo to the vehicle and sets a new 4 speed automatic transmission

Extension 3:

3.1) Business is expanding and there are many sales offices now. Aggregating the orders for the assembly line in a single file has become very time consuming. **Instead the customer now wants your program to be able to read from multiple files.**

3.2) A new Electric engine was developed and now all vehicles can be equipped with it. Electric vehicles do NOT have transmissions. The electric engine does not have displacement and in complies with the highest ecological standard as it does not pollute the environment. The output power is 535 kW.

Extension 4:

4.1) With the increased sales our car manufacturer wants to speed up the manufacturing process. They want to start a second instance of the assembly line in the same factory. The second instance will work in parallel to the first and share the same persistent storage.

4.2) Until now your software was creating vehicles instantly. This has confused the customer, because they got wrong impression about the time that it takes to manufacture the cars. To make things more realistic they have requested that you add a delay in process so that it takes

at least one second to create a vehicle and on each assembly line the cars have to be created one by one.