Project 1

Create an abstract class *Car*, including all fields and methods necessary for its proper functionality. Then create at least 3 inherit classes describing cars (i.e brands or types), based on this abstract class. Each of them should contain some unique fields describing its specific features.

Specific cars, like Volvo, Hundai or Skoda will be produced by independent factories (dedicated Thread for each). Each factory will get a task to produce 1000 new cars, but they can make new ones only if there is a free space at a parking area. Parking area is an integral part of each factory, and its s space is for 2 cars. It should be coded as your own implementation of generic list that will allow to store any random object, including these cars.

A truck cycles between factories (dedicated Thread), taking exactly 2 cars from each factory and carrying them to the customs parking, where they wait to be picked up by ships (implementing ship is not a part of this project). Each and every car remaining on customs parking is expected to have a unique id number, that should be checked (with Comparable interface) every time a new car is added. If you find a duplicate, an exception should be handled.

Both the truck and all 3 factories are working concurrent and with a max efficiency.

All the operations of picking up cars from factories and leaving them at customs should be logged into a file *log.txt*. Log format should be like: <factory car id number> <brand><left/taken>