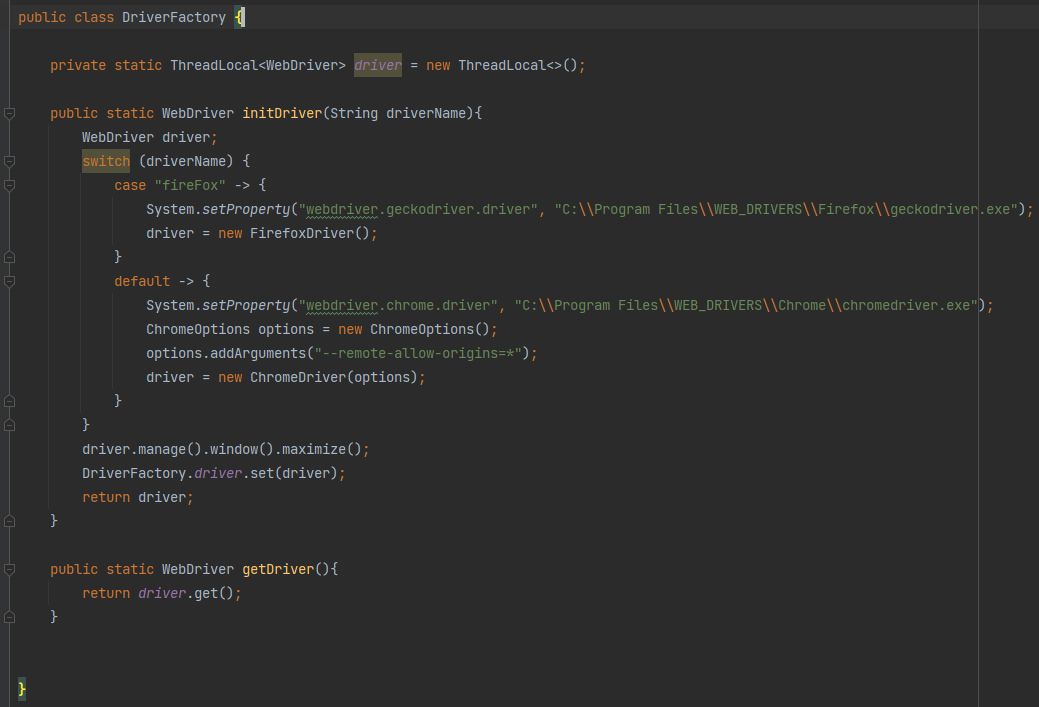
It helps us to instantiate a class (создавать экземпляр) and then inject the objects from that class into several other classes wherever required <- - all of this can be done by Dependency injection and not manually.

For example:



In that class we use static keyword in order to use the driver object into multiple classes. And in order to use parallel execution we need to use ThreadLocal that gives a copy of the driver variable to every thread. With pico container all of these can be avoided. The same class might look like this. Only the method that returns the driver object. No class variable at all!.



The only job that picocontainer does is to instantiate the objects (создавать экземпляр). So we just need to give some classes to the Pico container. It will create one instance of each class and it will hold these instance for every scenario. So every scenario will get its own instance of that class. So if this class has got several objects, for example: the driver object, or any of domain objects like product, billing details. So every scenario will get separate instance of all these objects. So in this way even if we run our scenario in parallel execution there will not be any conflict between the objects, because each scenario will have its own instance of the objects.

To use a pickle container we need just to add dependency to POM, no need for any of annotations.

<dependency>

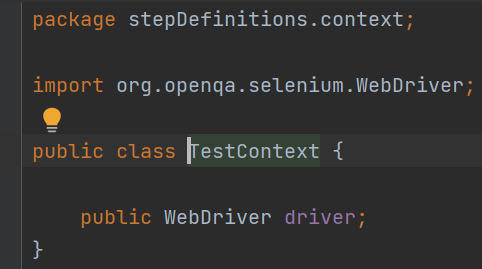
<groupId>io.cucumber</groupId>

<artifactId>cucumber-picocontainer</artifactId>

<version>7.14.0</version>

</dependency>

So first! create package and class in it TestContext



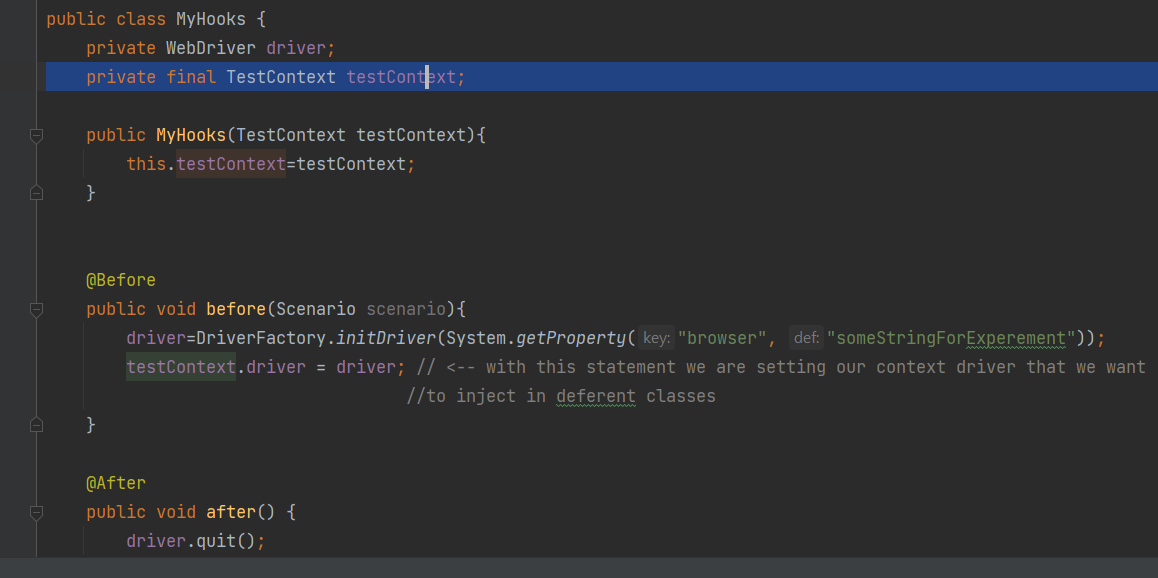
than in a class MyHooks we need to create :

1)-class variable of the class we just created TestContext testContext.

2)-create constructor (for MyHooks class) that takes as arguments object of a class we created TestContext testContext

3)-since in a created class TestContext exists only one class variable (it might be as many as needed) - Webdriver driver, than object of that class will have it as a field

4)-we need to have - Webdriver driver as a class variable (it's not important, important to initialize the variable testContext.driver), when we initialize that object we can assign the value to the class object that we created (TestContext ) since it has the same object as a field.



Then we can create object of the class we created (TestContext ) everywhere we need it.

for that :

1)create -> private WebDriver driver; as an instance of the class in which we want to use it.

2)-create a constructor that takes as arguments the object of the class we created before. (TestContext )

3)-since the object of the class that we created before has a field of Webdriver already assigned, we need to initiate the class variable with that field/value in the constructor.

And then use class variable as usual, in case of parallel execution cucumber will create an instance for every thread independently. COOOOL!!!

