Homework4

Programming Software Systems 2

March 30, 2021

Deadline is April 13, 2021, 23:59 (MSK)

.

The task is to implement a backend for the WEndex.Taxi .

Requirements:

- Implement two gateways. Passengers (passengers mobile app) should communicate with the system only through PassengerGateway class. Drivers (drivers mobile app) should communicate with the system only through DriverGateway class.
- A Passenger has name, rating, order history, payment methods, pinned addresses. He can
  - login, see order history, see and update payment methods, see and update pinned addresses.
  - select any two addresses (from, to) and carType to check the time and price of the ride. There are 4 car types: Economy, Comfort, ComfortPlus, Business.
  - order the ride if he agree with conditions (there is possibility to change default payment method for this particular ride).
  - ask for the current coordinates of the car during the ride.
  - see the ride in the order history after the end of the ride and ask for the bill of that ride.
- A driver has name, rating, order history, one car, status(is working or not, in ride or not). He can login, see order history, see a car, update status, check available orders, accept or decline it.
- A Car has model, carType, current coordinates, the color and the number. If the type is Comfort it has count of freeBottleOfwater (it randomly decreases during rides). If the type is Business it has the method parkRightInFrontOfTheEntrance. The order has time and two addresses.

- Database. Use any. The simplest way - just a file. But you should not miss any data from launch to launch.
- Show work of the system in the main function.
- If something was not mentioned - it means on your choice.
- All common requirements from previous tasks (readme, readability, extensibility, DRY, OOP etc).

Not Required:

- To implement honest matching algorithm.
- To implement real coordinate system and addresses.
- To use threads (at least in this work). You can do everything in one thread.

Some suggestions (the same):

Ideally - all results should be in the PullRequest(PR) in a separate branch in the private repository on github. In such case put only the link to the repo in the moodle. (Do not forget to add TA to the repo. Mike's login is **l1va**, Shokhista's - **ergashevashokhista**)

Merge PR only after we approve it.

Or you can upload zip archive to moodle if you can't manage with git.

Do not leave the task for the last day. You can meet difficulties that you will not manage to solve in 30 minutes.