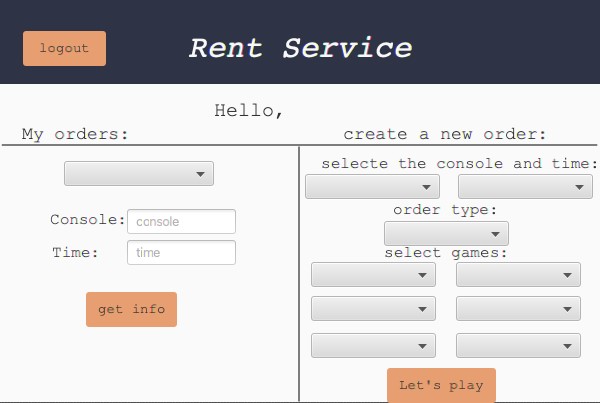
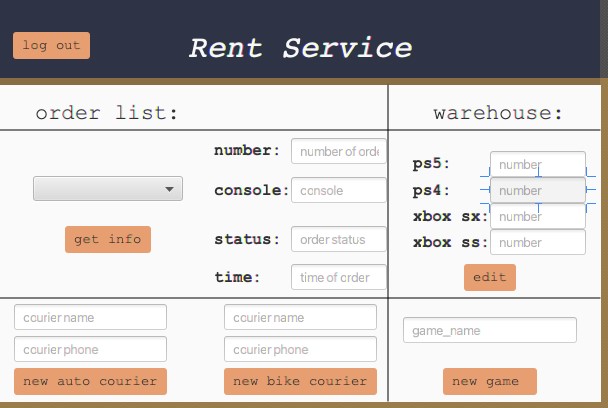
Rent manager

Quick guide to using the program:

to log in to the admin window, enter "admin"in the password and login field. First, you need to create games in the admin window and enter the data of your couriers.



Couriers can log in to the app if they enter their name in the login field and their phone number in the password field.

Attention!!!

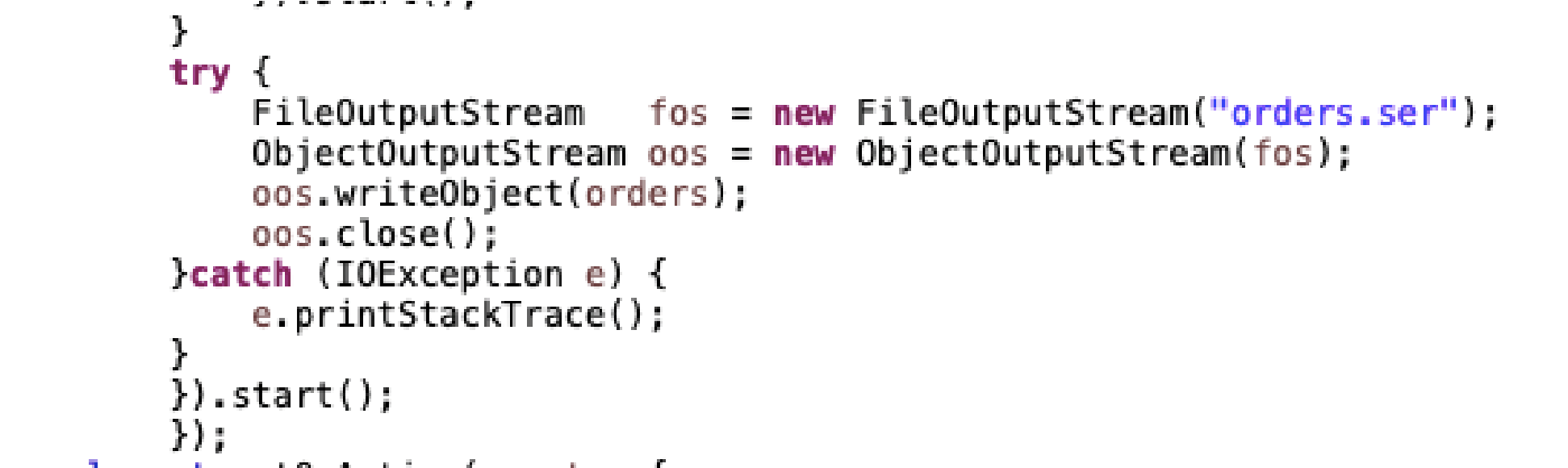
The courier can ONLY log in if he has an active order.

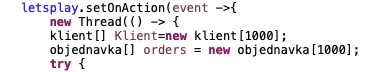
For regular customers, the sign-in system works as usual.

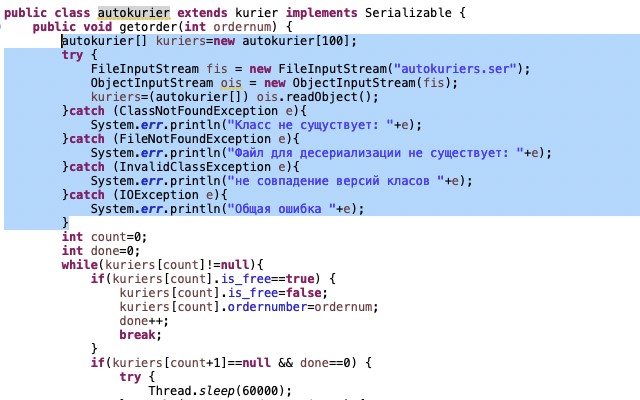
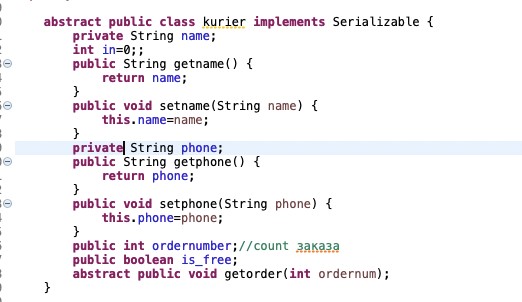
Additional criteria:

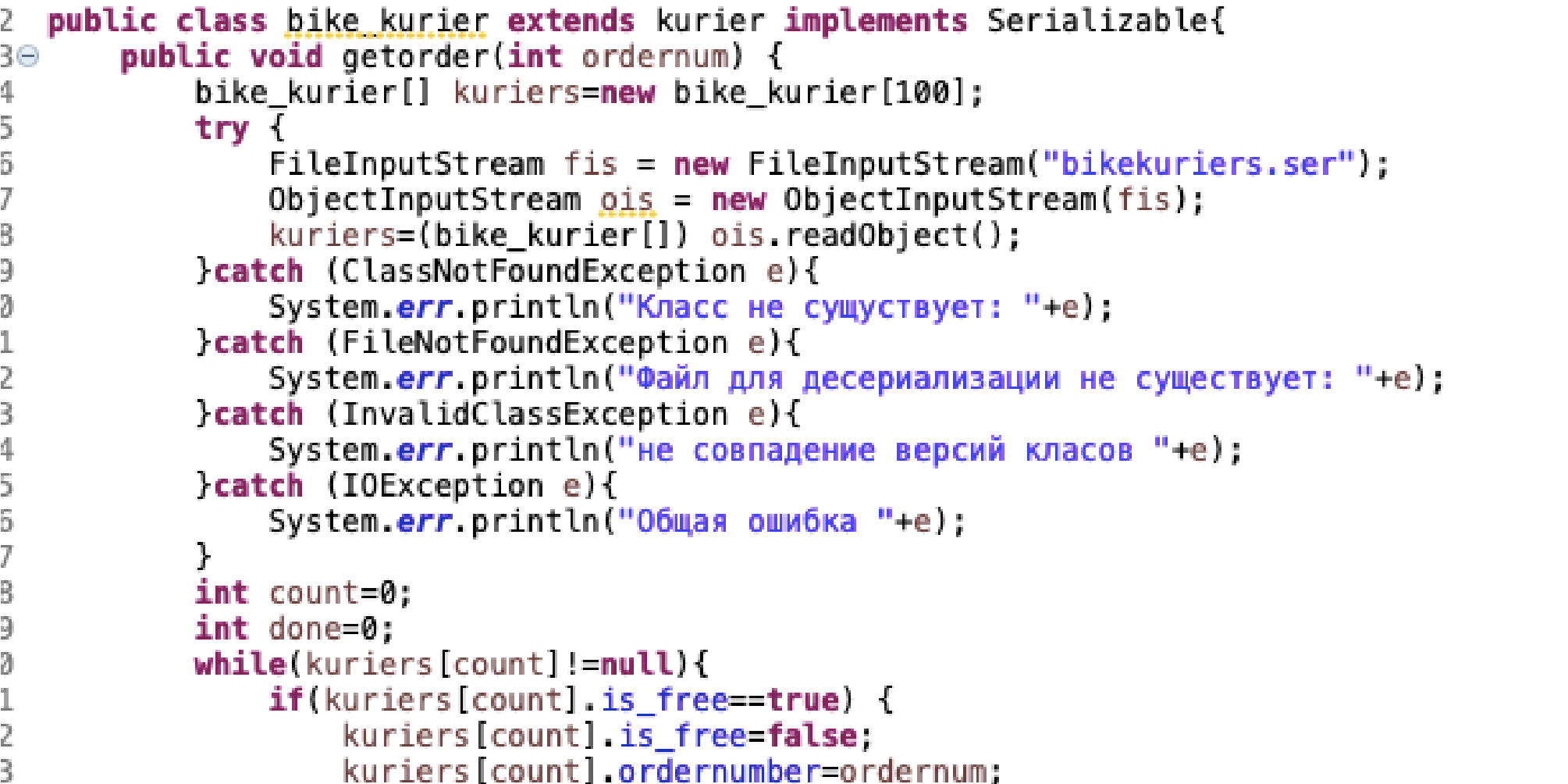
The separation of the user interface from the application logic is implemented using controllers that control each individual application window.



I use serialization multiple times to store user data, and warehouse data. In order for the program not to hang during serialization, data is written in a separate thread, which, for convenience, is output in a lambda expression.



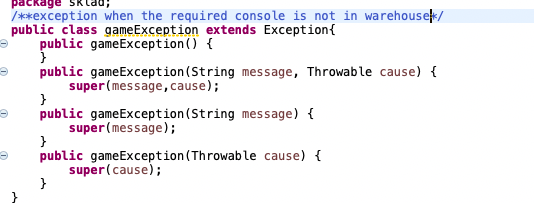


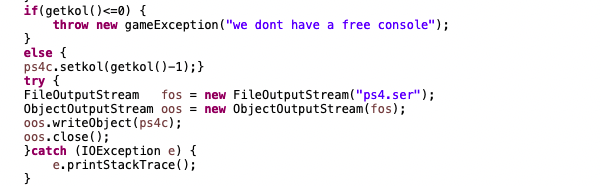
I use inheritance for the courier on car and courier on bike classes and the console class.Polymorphism is used to search for a free courier, if the user wants fast delivery, the search for a car courier is performed, in the case of a regular one - by bicycle.

In the case of consoles, polymorphism is used to implement a warehouse, when when calling getconsole (), 1 console of the desired type is taken from the warehouse file

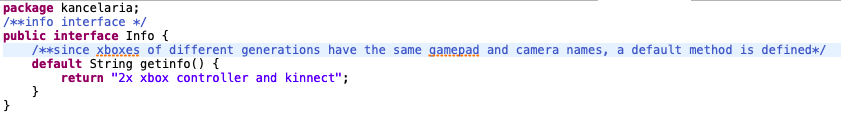
I am implementing aggregation using auto classes in the courier inheritance branch, and complectation classes in the games inheritance branch

I have my own exception class that handles the exception when the required consoles to order are not enough in the warehouse





I also implement the 'Info' interface method that returns data as a string. It is used in the inheritance branch of couriers and games, but since xboxes have the same result for using GetInfo, a standard method is defined for them to reduce the length of the code



incapsulation is implemented using private modifiers and creating getters and setters for each variable

