Activity 5

Online Multiple-Choice Exams

Authors:

Francisco Javier Roig Gregorio - 47433543E

Jose Miguel Avellana López - 18060891G

Course 2020 – 2021

Profesor

The Teacher's class has a *professorThread* that is used to obtain the instruction input, in our case we will use it to determine when to finish the exam.

The following instructions are executed in the main thread:

1. The csv file containing the questions, their options and the correct answer is uploaded.
2. All students are expected to have connected to the exam room.
3. Begin the exam by notifying each of the students.
4. All questions and their corresponding options are sent to each of the students.
5. A thread is initialized to control the students that are connected.
6. A loop is executed that ends when all responses are received from all students.
7. The exam is over.

The professor implementation has an *Exam* object, which has an *Questions* list and a variable to know if the exam has been initialized.

With it, the professor can know if an answer is true or not.

The professor will create questions introducing all parameters, but when he will go to send it to his students, he will create a new question object for each other, only with the question and the set of answers.

Student

The Student is initialized by reading an input that will determine his ID, once started he will join the exam. If this has already started, you will receive a notification that will indicate it, otherwise it will continue with its normal execution.

You will start receiving the questions and your choices from the teacher and will save them in a queue. This queue will be emptied as the questions are answered.

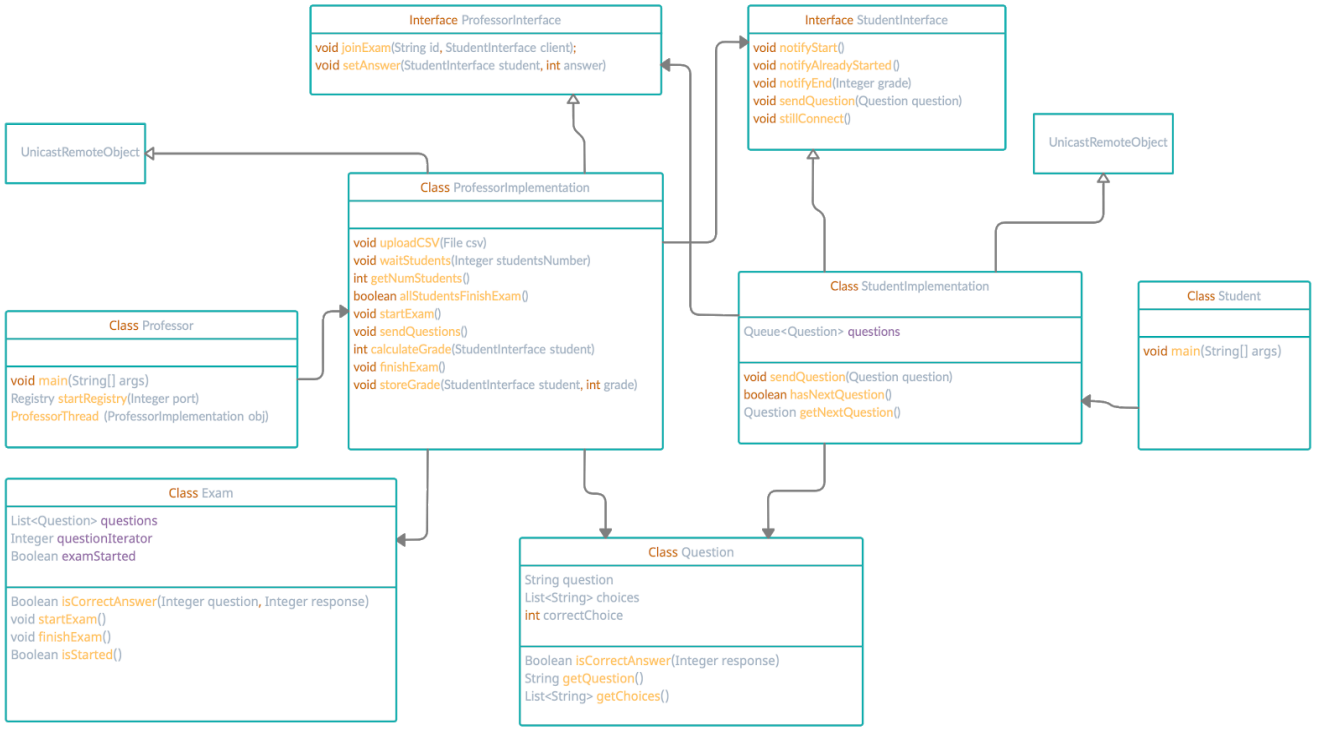
And finally, each time a question is answered, the chosen option will be sent to the teacher.

Common

The *Question* object is common. It is composed of the question, a set of possible answers and the number of the correct one.

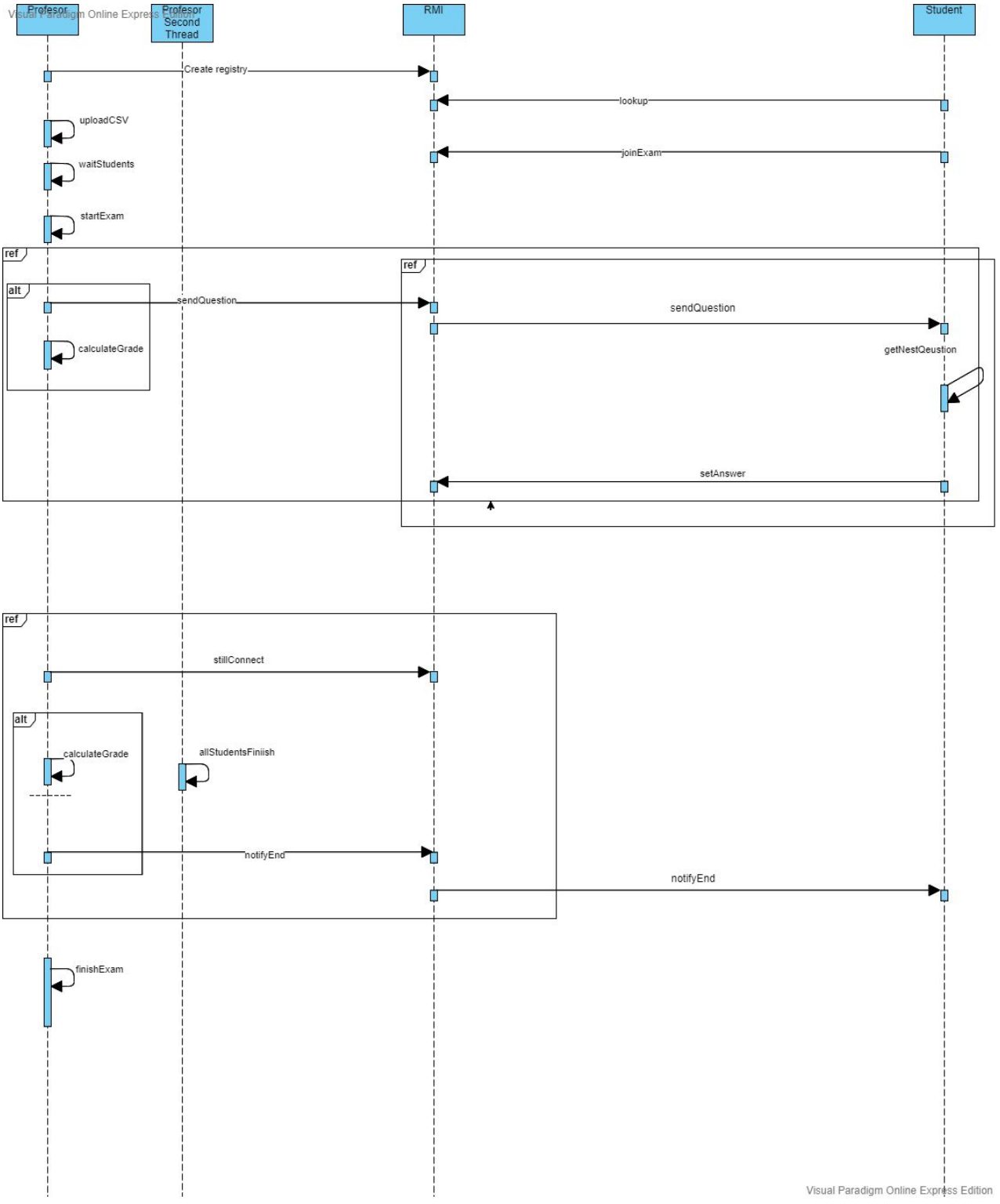
It is a serializable element that allows us to pass the information using it as a data structure.

UML class diagram



UML sequence diagram

Perdona, no nos fue el editor.



Github Repo:

[Javigoo/DistributedQuest-JavaRMI (github.com)]