# TS Advanced: Regular Exam - 18.01.20

# Problem 2. Qualification

**export** **class** Quali {  
 *//* ***TODO: implement this class...***  
}

### Your Task

### Write a Quali class, which supports the described functionality below. Firstly, import the QualiData custom type and the Racers class.

### Functionality

#### constructor()

Receives **no** parameters at initialization of the class, butshould have these **2** properties:

* **times** - empty array of **QualiData** type
* **finalPositions** - empty array of **QualiData** type

#### fastestLap({id}, {lapTime})

Both the id and the lapTime are **numbers**

* Check if the given **id** corresponds to a racer in the **racers** array (from the imported Racers class) and if he has a signed contract, if not throw a new error:

**"There is no such racer on the grid or has no contract with a team"**

* Otherwise push an object with properties the arguments passed to the method in the **times** array and sort it every time, in ascending order by the lap time, a new lap is completed.

#### standings()

* Take the lap times from the times array and push an object with the driver id and his corresponding position into the **finalPositions** array. Note that the position must start from 1. After than return a message:

**"The fastest driver was {firstName} {lastName} with car {id}!"**

### Examples

This is an example how the code is **intended to be used**:

|  |
| --- |
| Sample code usage |
| **let q = new Quali();**  **q.fastestLap(22, 1.24);**  **q.fastestLap(44, 1.22);**  **q.fastestLap(5, 1.25);**  **q.fastestLap(33, 1.23);**  **console.log(q.standings());** |
| Corresponding output |
| The fastest driver was Lewis Hamilton with car 44! |