**OOP**

**DEADLINE:** 24.05.2020

## FOLDER STRUCTURE

|  |  |
| --- | --- |
| FL13\_HW14/\*  └─ homework/\*  └─ index.html\*  └─ .eslintrc.js  └─ js\*  └─ app.js\*  └─ data.js\* | \*   – required |

|  |  |
| --- | --- |
|  |  |

## TASK

### 1. You should create a function constructor ‘**Student’** which you can call with two arguments name and email.

### **Name**, **email** and **homeworkResults** fields should be hidden from user.

### Instance of Student should provide the following methods to get or change it’s state:

### **getName**(): returns student’s name.

### **getEmail**(): returns student’s email.

### **addHomeworkResult**(topic, success): you can call this method with 2 arguments: topic(string) and success(boolean). This method should add new element to student’s homeworkResults property.

A screenshot of text

Description automatically generated

### 2. You should create a function constructor ‘**FrontendLab’** which you can call with two arguments students and failedLimit.

### Instance of FrontendLab should have 2 property fields: **failedHomeworksLimit** and **studentsList**. This fields should be hidden from user.

### Instance of Student should provide the following methods to get or change it’s state.

### **printStudentsList**(): this method logs to console list of students with their homeworks results.

A screenshot of a cell phone

Description automatically generated

### **addHomeworkResults**(homeworkResults): this method can be called with argument homeworkResult, object with 2 fields: topic(string) and results(array of objects with 2 fields: email(string) and success(boolean)). This method should update all student objects in FrontendLab studentsList.

### **printStudentsEligibleForTest**(): this method should log to console list of students who didn’t fail more homework, than allowed failedLimit.

A screenshot of a cell phone

Description automatically generated

## BEFORE SUBMIT

### Remove all unnecessary files that you might have included by mistake

### Verify that all functionality is implemented according to requirements

### Make sure you code is well-formatted, and validated via validator (w3org Markup Validation Service)

### Add comments if the code is difficult to understand

### Fix warnings/errors in the browser console

### Verify that the name of the folders and files meet the requirements

### Make sure there are no errors/warnings in the browser console

### Run the linter and fix all warnings and errors.

**HOW TO**

### Use linter :

### In order to use npm package manager you should install nodejs (https://nodejs.org/ )

### Install eslint to check your code (npm install -g eslint)

### - open a terminal(or cmd)

### - run eslint (i.e. eslint ./js/task1.js)

### Code should be without ‘errors’

## SUBMIT

### The folder should be uploaded to github repository ‘**FL-13**’ into **master** branch