OOP

**DEADLINE:** 25/03/2018

## FOLDER STRUCTURE

homework\_03/

├─ js/

├─ main.js

└─ index.html

## TASK

1. You should create two classes **Employee** and **Company**.
2. Constructor of **Company** class should expect three arguments:
   1. **name** - Name of company.
   2. **owner** - Owner of company.
   3. **maxCount** - Max count of employees who could work in company.
3. Constructor of **Employee** class should expect four arguments:
   1. **name** - Name of employee.
   2. **primarySkill** - Primary skill.
   3. **age** - Age of employee.
   4. **salary** - Salary of employee.
4. **Company** class should have at least following methods:
   1. **addNewEmployee** - add new employee to list of the company( If there no free place for new employee, you should remove employee who has the lowest salary and add new employee).
   2. **removeEmployee** - remove employee by id(it can be index in array).
   3. **getAvarageSalary** - return average salary of employees.
   4. **getEmployees** - return all employees. (employees field should be private).
   5. **getFormattedListOfEmployees** - return list of employees in next format `${name} - works in ${companyName} ${timeInCompany} seconds`
   6. **getAvarageAge** - return average age of employees.
   7. **getHistory** - return history log(you should have private field **\_logs** where save all information about actions).
5. **Employee** class should have at least following methods:
   1. **getSalary** - return salary of employee.
   2. **setSalary** - sets salary of employee(cannot set smaller salary than employee has now).
   3. **getWorkTimeInSeconds** - return work time in seconds (should calculate time which employee works in all company).
   4. **hire** - sets current name of company.
   5. **fire** - remove name of company.
   6. **getHistory** - return history log(All actions fire/hire should be save information message in private field).
6. All functions should contain assertions that prevent non-logical behaviour(example: set to salary only finite number)
7. There should be some script that demonstrates work of all functions.

## RESTRICTIONS

* Do not use ES6 classes
* no need to use any of build systems(grunt, gulp, webpack).

## CODE

let artem = new Employee({name: "Artem", age: 15, salary: 1000, primarySkill: "UX"});

let vova = new Employee({name: "Vova", age: 16, salary: 2000, primarySkill: "BE"});

let vasyl = new Employee({name: "Vasyl", age: 25, salary: 1000, primarySkill: "FE"});

let ivan = new Employee({name: "Ivan", age: 35, salary: 5000, primarySkill: "FE"});

let orest = new Employee({name: "Orest", age: 29, salary: 300, primarySkill: "AT"});

let anton = new Employee({name: "Anton", age: 19, salary: 500, primarySkill: "Manager"});

let epam = new Company({name: "Epam", owner: "Arkadii", maxCompanySize: 5});

epam.addNewEmployee(artem);

epam.addNewEmployee(vova);

epam.addNewEmployee(vasyl);

epam.addNewEmployee(ivan);

epam.addNewEmployee(orest);

epam.addNewEmployee(anton);

console.log(epam.getHistory());

/\*

"Epam was created in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time)"

"Artem starts working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

"Vova starts working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

"Vasyl starts working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

"Ivan starts working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

"Orest starts working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

"Orest ends working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

"Anton starts working at Epam in Mon Mar 12 2018 07:41:50 GMT+0200 (FLE Standard Time);"

\*/

epam.removeEmployee(2);

console.log(vasyl.getHistory());

/\*

"Vasyl is hired to Epam in Mon Mar 12 2018 07:45:55 GMT+0200 (FLE Standard Time)"

"Vasyl is fired from Epam in Mon Mar 12 2018 07:45:55 GMT+0200 (FLE Standard Time)"

\*/

console.log(epam.getAvarageSalary()); // -> 2125

console.log(epam.getAvarageAge()); // -> 21.25

epam.addNewEmployee(5,6,9,5); // -> Please try to add Employee instance

setTimeout(() => {

epam.removeEmployee(1);

console.log(artem.getWorkTimeInSeconds()); // -> 5.5744444444444445

}, 5000);

vova.setSalary(900);

vova.setSalary(2200);

console.log(vova.getHistory());

/\*

"Vova is hired to Epam in Mon Mar 12 2018 08:08:48 GMT+0200 (FLE Standard Time)"

"try to change salary from 2000 to 900"

"change salary from 2000 to 2200"

\*/

## BEFORE SUBMIT

* check files structure;
* check code structure(code style);

## USEFUL LINKS

* <http://eloquentjavascript.net/1st_edition/chapter8.html>
* <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_JS>