**Problem 3. University**

Exam problems for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/trainings/3011/js-advanced-september-2020). Submit your solutions in the SoftUni Judge system at <https://judge.softuni.bg/Contests/Compete/Index/2714#2>

Implement the following classes: **Person, Teacher, Student.**

**Submission**

Submit your **solveClasses** function with the implementation of the three classes**.**

**Class Person**

**constructor ( firstName, lastName )**

Should have these **3** properties:

* **firstName – string;**
* **lastName – string;**
* **problems – array;**

**toString ()**

This **method** returns a string as follows:

**"{firstName} {lastName} is part of SoftUni community now!"**

**Class Teacher**

Class **Teacher** inherits class **Person**.

**constructor ( firstName, lastName)**

Should have these **2** properties:

* **firstName – string;**
* **lastname – string;**

**createProblem ( id, difficulty )**

This **method** creates a Problem (object) with the keys: **id** and **difficulty.** After that the problem is created it should be **added** to the **problems array**.

The method **returns** the **problems** **array** at the end.

**getProblems()**

This method **returns** the **array** with all the created **problems**.

**showProblemSolution ( id)**

This method should try to **find** the problem **by the given id.** If the given **id is NOT found** the method should **throw an Error**:

**"Problem with id { id } not found."**

Else the **difficulty** of this problem should be **reduced by 1** and the **problem** should be **returned**.

**Class Student**

Class **Student** inherits class **Person**.

**constructor( firstName, lastName, graduationCredits, problems )**

Should have these **6** properties:

* **firstName – string;**
* **lastName – string;**
* **graduationCredits - number;**
* **myCredits = 0;**
* **solvedProblems = [];**
* **problems – array of object**

**solveProblem(id)**

This method should try to **find** the problem **by the given id.** If the given **id is NOT found** the method should **throw an Error**:

**"Problem with id { id } not found."**

Else the student's credits (**myCredits**) should be **increased** by the **difficulty** **of the problem**.

The **problem** should be **added** to the array with the **solvedProblems.**

If the problem already exists in this array **(check for the id)** you should **NOT** increase the credits of the student.

At the end the **method** **returns** the student's **credits**.

**graduate()**

This method checks if the student **has enough credits for graduation** (**myCredits** >= **graduationCredits**). If so the method **returns**:

**"{firstName} {lastName} has graduated succesfully."**

If the student has **NOT** enough credits the method calculates **the needed credits** and **returns**:

**"{firstName} {lastName}, you need { neededCredits } credits to graduate."**

**Examples**

|  |
| --- |
| **Sample code usage** |
| **const classes = solveClasses();**  **const teacher = new classes.Teacher("Ivailo", "Papazov");**  **teacher.createProblem('as442df', 5);**  **console.log(teacher.problems);**  **teacher.createProblem('ffff44', 15);**  **console.log(teacher.problems);**  **teacher.showProblemSolution('as442df');**  **console.log(teacher.problems);**  **----------------------------------------------------------------------------**  **const classes = solveClasses();**  **const student = new classes.Student("Pesho", "Petrov", 23, [{id: '111', difficulty: 5}, {id: '222', difficulty: 15}]);**  **student.solveProblem('111');**  **console.log(student.myCredits);**  **console.log(student.graduate());**  **student.solveProblem('222');**  **console.log(student.solvedProblems);**  **console.log(student.graduate());** |
| **Corresponding output** |
| **[{id: 'as442df', difficulty: 5}]**  **[{id: 'as442df', difficulty: 5}, {id: 'ffff44', difficulty: 15}]**  **[{id: 'as442df', difficulty: 4}, {id: 'ffff44', difficulty: 15}]**  **----------------------------------------------------------------------------**  **5**  **Pesho Petrov, you need 18 credits to graduate.**  **[{id: '111', difficulty: 5}, {id: '222', difficulty: 15}]**  **Pesho Petrov, you need 3 credits to graduate.** |