**Problem 2. Employees**

Exam problems for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/courses/javascript-advanced). Submit your solutions in the SoftUni Judge system at <https://judge.softuni.bg/Contests/Compete/Index/2590#1>

Implement the following classes: **Developer, Junior, Senior.**

**Submission**

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

**Class Developer**

**constructor ( firstName, lastName )**

Should have these **6** properties:

* **firstName – string;**
* **lastName – string;**
* **baseSalary - 1000;**
* **tasks – array;**
* **experience - 0;**

**addTask ( id, taskName, priority )**

This **method** will receive **id (number), name (string) and priority (string)**. Create a new **task object** and add it to the **tasks** array. **Return** a message:

**"Task id {id}, with {priority} priority, has been added."**

If the task has a **"high"** priority, add it **as the first task**, else - add it **as the last**.

Note: The task **id** will **always be unique**.

**doTask()**

This **method** removes **the newest task** with **the highest priority** and returns the **task's name**.  
If there are no tasks, return:

**"{first name}, you have finished all your tasks. You can rest now."**

**getSalary()**

This **method** should **return:**

**"{firstName} {lastName} has a salary of: {salary}"**

**reviewTasks()**

This **method** should **return** all of the **incompleted tasks** in the format**:**

**"Tasks, that need to be completed:**

**{id}: {name} - {priority}**

**(...)"**

**Class Junior**

Class **Junior** inherits class **Developer**.

**constructor( firstName, lastName, bonus, experience )**

Should have these **6** properties:

* **firstName – string;**
* **lastname – string;**
* **baseSalary – 1000 + bonus;**
* **tasks – array;**
* **experience - number;**

**learn( years )**

This **method** should increase **the years of experience** of the junior developer

**Class Senior**

Class **Senior** inherits class **Developer**.

**constructor( firstName, lastName, bonus, experience )**

Should have these **6** properties:

* **firstName – string;**
* **lastname – string;**
* **baseSalary - 1000 + bonus;**
* **tasks – array;**
* **experience - number + 5;**

**changeTaskPriority(taskId)**

This **method** should first change the task's priority (**"high"** becomes **"low"** / **"low"** becomes **"high"**) and then the **task should be moved** on the **first** place of the **tasks array** if the priority is **high**, and on the **last** place if the **priority** is **low**. At the end the method should **return the task**.

**Examples**

|  |
| --- |
| **Sample code usage** |
| **let classes = solveClasses();**  **const developer = new classes.Developer("George", "Joestar");**  **console.log(developer.addTask(1, "Inspect bug", "low"));**  **console.log(developer.addTask(2, "Update repository", "high"));**  **console.log(developer.reviewTasks());**  **console.log(developer.getSalary());**  **----------------------------------------------------------------------------**  **const junior = new classes.Junior("Jonathan", "Joestar", 200, 2);**  **console.log(junior.getSalary());**  **----------------------------------------------------------------------------**  **const senior = new classes.Senior("Joseph", "Joestar", 200, 2);**  **senior.addTask(1, "Create functionality", "low");**  **senior.addTask(2, "Update functionality", "high");**  **console.log(senior.changeTaskPriority(1)["priority"]);** |
| **Corresponding output** |
| **Task id 1, with low priority, has been added.**  **Task id 2, with high priority, has been added.**  **Tasks, that need to be completed:**  **2: Update repository - high**  **1: Inspect bug - low**  **George Joestar has a salary of: 1000**  **----------------------------------------------------------------------------**  **Jonathan Joestar has a salary of: 1200**  **----------------------------------------------------------------------------**  **high** |