# JS Advanced - Retake Exam: 09.12.2020

Exam problems for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/trainings/3011/js-advanced-september-2020/internal). Submit your solutions in the SoftUni Judge system at <https://judge.softuni.bg/Contests/Compete/Index/2714#1>.

## Problem 2. Vacation

|  |
| --- |
| **class** Vacation {  *//* ***TODO: implement this class...*** } |

**Your Task**

Write a **class Vacation**, which implements the following functionality:

**Functionality**

#### constructor(organizer, destination, budget)

Receives **2** parameters at initialization of the class (**organizer, destination** and **budget**).

Should have at least these **4 properties**:

* **organizer** – property of type string;
* **destination** – property of type string;
* **budget** – property of type number;
* **kids**– initially an empty object;

#### registerChild(name, grade, budget)

Stores **all kids** into the **kid's property** by their **grade**. Every grade corresponds to **array** of all **kids** in there by following format: **{name}-{budget}**

* **Check** if the current kid’s budget is **enough** for the trip. If it is not, it should **return** the following string:

**'{name}'s money is not enough to go on vacation to {destination}.'**

* Otherwise we **add it** if it is not already recorded in and **return the current grade.**
* If current kid is already into that **grade** the method should **return** the following **string**:

**'{name} is already in the list for this {destination} vacation.'**

#### removeChild(name, grade)

**Removes** a kid from the **array** of already enrolled kids for this trip if the current kid **exists** in there of course…

* If the **name** of the **current kid** **exists** in the current grade, we **remove** him and **return** the **current** **grade.**
* If the given kid name does **not exist** in the given grade, we should **return** the following **string**:

**'We couldn't find {name} in {grade} grade.'**

#### toString()

**Returns** all kids from the **kid's** property **sorted** in **ascending** order by their **grade** into the following format:

**`{organizer} will take {numberOfChildren} children on trip to {destination}  
`Grade: {currentGrade}`  
{currentKidNumber}. {kid}`  
…  
  
`Grade: {nextGrade}`  
{currentKidNumber}. {kid}`  
…**

The new line ("**\n"**) after every kid or **grade is at the end**.

* If there **are currently no kids** for the current trip, the kids property is **empty** and you should **return** the following **string**:

**`No children are enrolled for the trip and the organization of ${this.organizer} falls out...**` **Check the example below for more clarity**.

#### numberOfChildren()

**Returns** the current **count** of **all kids** into the **kid's** property.

### Notes

**Names of all functions must be exactly the same as in the description and examples!**

### Examples

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

|  |
| --- |
| **Sample code usage** |
| **let *vacation* = new Vacation('Mr Pesho', 'San diego', 2000);**  ***console***.log(***vacation***.registerChild(**'Gosho'**, 5, 2000));  ***console***.log(***vacation***.registerChild(**'Lilly'**, 6, 2100));  ***console***.log(***vacation***.registerChild(**'Pesho'**, 6, 2400));  ***console***.log(***vacation***.registerChild(**'Gosho'**, 5, 2000));  ***console***.log(***vacation***.registerChild(**'Tanya'**, 5, 6000));  ***console***.log(***vacation***.registerChild(**'Mitko'**, 10, 1590)); |

|  |
| --- |
| **Corresponding output** |
| [ 'Gosho-2000' ]  [ 'Lilly-2100' ]  [ 'Lilly-2100', 'Pesho-2400' ]  Gosho is already in the list for this San diego vacation.  [ 'Gosho-2000', 'Tanya-6000' ]  Mitko's money is not enough to go on vacation to San diego. |

|  |
| --- |
| **Sample code usage** |
| **let *vacation* = new Vacation('Mr Pesho', 'San diego', 2000);**  ***vacation***.registerChild(**'Gosho'**, 5, 2000);  ***vacation***.registerChild(**'Lilly'**, 6, 2100);  ***console***.log(***vacation***.removeChild(**'Gosho'**, 9));  ***vacation***.registerChild(**'Pesho'**, 6, 2400);  ***vacation***.registerChild(**'Gosho'**, 5, 2000);  ***console***.log(***vacation***.removeChild(**'Lilly'**, 6));  ***console***.log(***vacation***.registerChild(**'Tanya'**, 5, 6000)) |
| **Corresponding output** |
| We couldn't find Gosho in 9 grade.  [ 'Pesho-2400' ]  [ 'Gosho-2000', 'Tanya-6000' ] |

|  |
| --- |
| **Sample code usage** |
| **let *vacation* = new Vacation('Miss Elizabeth', 'Dubai', 2000);**  ***vacation***.registerChild(**'Gosho'**, 5, 3000);  ***vacation***.registerChild(**'Lilly'**, 6, 1500);  ***vacation***.registerChild(**'Pesho'**, 7, 4000);  ***vacation***.registerChild(**'Tanya'**, 5, 5000);  ***vacation***.registerChild(**'Mitko'**, 10, 5500)  ***console***.log(***vacation***.toString()); |
| **Corresponding output** |
| Miss Elizabeth will take 4 children on trip to Dubai  Grade: 5  1. Gosho-3000  2. Tanya-5000  Grade: 7  1. Pesho-4000  Grade: 10  1. Mitko-5500 |