# JS Advanced: Exam 13 November 2016

Problems for exam preparation 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/699/>.

# Problem 2. Sumator Class (Unit Testing)

You are given the following **JavaScript class**:

|  |
| --- |
| sumator.js |
| **class** Sumator {  constructor() {  **this**.**data** = [];  }  add(item) {  **this**.**data**.push(item);  }  sumNums() {  **let** sum = 0;  **for** (**let** item **of this**.**data**)  **if** (**typeof** (item) === **'number'**)  sum += item;  **return** sum;  }  removeByFilter(filterFunc) {  **this**.**data** = **this**.**data**.filter(x => !filterFunc(x));  }  toString() {  **if** (**this**.**data**.**length** > 0)  **return this**.**data**.join(**", "**);  **else  return '(empty)'**;  } } |

### Functionality

The above code defines a **class** that holds items (of **any** type). An **instance** of the class should support the following operations:

* Contains a property data that is initialized to an **empty** array.
* Functionadd(item) – **adds** the passed in **item** (of **any** type) to the data.
* FunctionsumNums()– **sums** only the **numbers** from the data and **returns** the sum. If there are **no** numbers stored, the function should return **zero**.
* FunctionremoveByFilter(filterFunc)– **filters** the data by a given **function**. All of the items that **match** the criteria should be **removed**.
* FunctiontoString()– **returns** a string, containing a list of all items from the data, joined with a **comma** and a **space**. If there are **no** items stored, it should **return** the string **"(empty)".**

### Examples

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

|  |  |  |
| --- | --- | --- |
| Sample code usage |  | Corresponding output |
| **let** list = **new** Sumator();  console.log(**`list = [**${list}**]`**); list.add(1); list.add(2); list.add(**"three"**); list.add(4); console.log(**`list = [**${list}**]`**); console.log(**"sum = "** + list.sumNums()); list.add(**"5.5"**); *// not a number!* list.add(7.7); console.log(**`list = [**${list}**]`**); console.log(**"sum = "** + list.sumNums()); list.removeByFilter(x => x % 2 === 0); console.log(**`list = [**${list}**]`**); console.log(**"sum = "** + list.sumNums()); | list = [(empty)]  list = [1, 2, three, 4]  sum = 7  list = [1, 2, three, 4, 5.5, 7.7]  sum = 14.7  list = [1, three, 5.5, 7.7]  sum = 8.7 |

### Your Task

Using **Mocha** and **Chai** write **JS unit tests** to test the entire functionality of the Sumator class. Make sure it is correctly defined as a class and instances of it have all the required functionality. You should have at least **7 test cases**. You may use the following code as a template:

|  |
| --- |
| describe(**"*TODO* …"**, **function**() {  ***it***(**"*TODO …*"**, **function**() {  *//* ***TODO:*** …  });  *//* ***TODO:*** …  }); |

### Submission

Submit your tests inside a describe() statement.