

Задача 1

Реализовать менеджер таймеров:

```
1  class TimersManager {
2      constructor() {
3          this.timers = [];
4      }
5
6      add() {}
7
8      remove() {}
9
10     start() {}
11
12     stop() {}
13
14     pause() {}
15
16     resume() {}
17 }
18
19 const manager = new TimersManager();
20
21 const t1 = {
22     name:      't1',
23     delay:     1000,
24     interval:  false,
25     job:       () => { console.log('t1') }
26 };
27
28 const t2 = {
29     name:      't2',
30     delay:     1000,
31     interval:  false,
32     job:       (a, b) => a + b
33 };
34
35 manager.add(t1);
36 manager.add(t2, 1, 2);
37 manager.start();
38 console.log(1);
39 manager.pause('t1');
```

1. Метод **add** должен добавлять таймер в очередь на выполнение. В качестве первого параметра этот метод принимает объект описывающий таймер, а все последующие

параметры передаются как аргументы для callback функции таймера.

2. Вызовы метода **add** можно соединять `manager.add(t1).add(t2, 1, 2);`
3. Метод **remove** должен остановить определённый таймер и удалить его из очереди.
4. Метод **start** должен запустить все таймеры на выполнение.
5. Метод **stop** должен остановить все таймеры.
6. Метод **pause** приостанавливает работу конкретного таймера.
7. Метод **resume** запускает работу конкретного таймера
8. Таймеры могут быть как одноразовыми (выполнить задачу через определённый промежуток времени), так и периодическими (выполнять задачу с определённым интервалом). Если **interval** = true — таймер периодический.

Обратите внимание!

1. TimeManager должен вызывать ошибку если поле **name** содержит неверный тип, отсутствует или пустая строка.
2. TimeManager должен вызывать ошибку если поле **delay** содержит неверный тип или отсутствует.
3. TimeManager должен вызывать ошибку если **delay** меньше **0** и больше **5000**.
4. TimeManager должен вызывать ошибку если поле **interval** содержит неверный тип или отсутствует.
5. TimeManager должен вызывать ошибку если поле **job** содержит неверный тип или отсутствует.
6. TimeManager должен вызывать ошибку если запустить метод **add** после старта.
7. TimeManager должен вызывать ошибку если попытаться добавить таймер с именем которое уже было добавлено.

Timer:

```
1 {  
2     name*:      String, // timer name  
3     delay*:     Number, // timer delay in ms  
4     interval*: Boolean, // is timer or interval  
5     job*:       Function, // timer job  
6 }
```