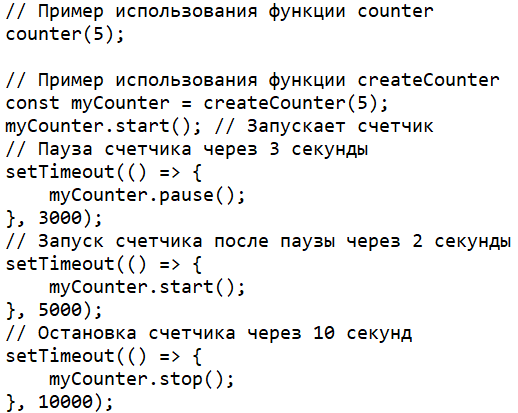
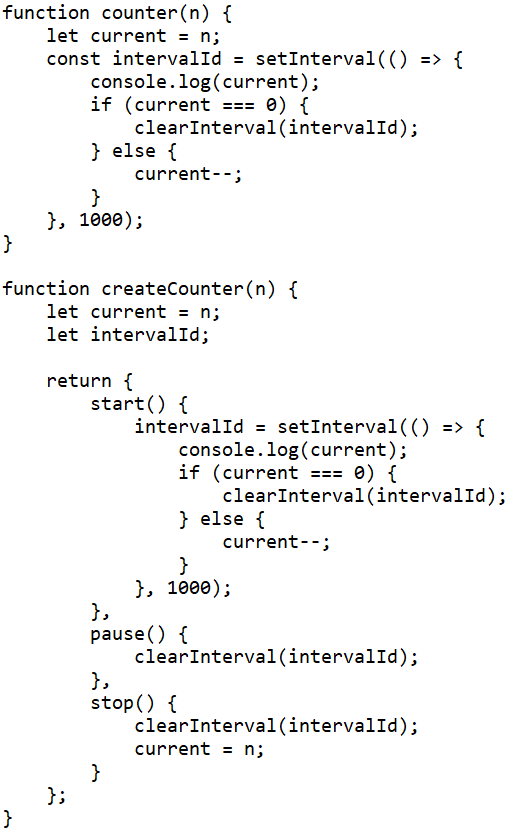
**Задание 1 (на таймеры)**

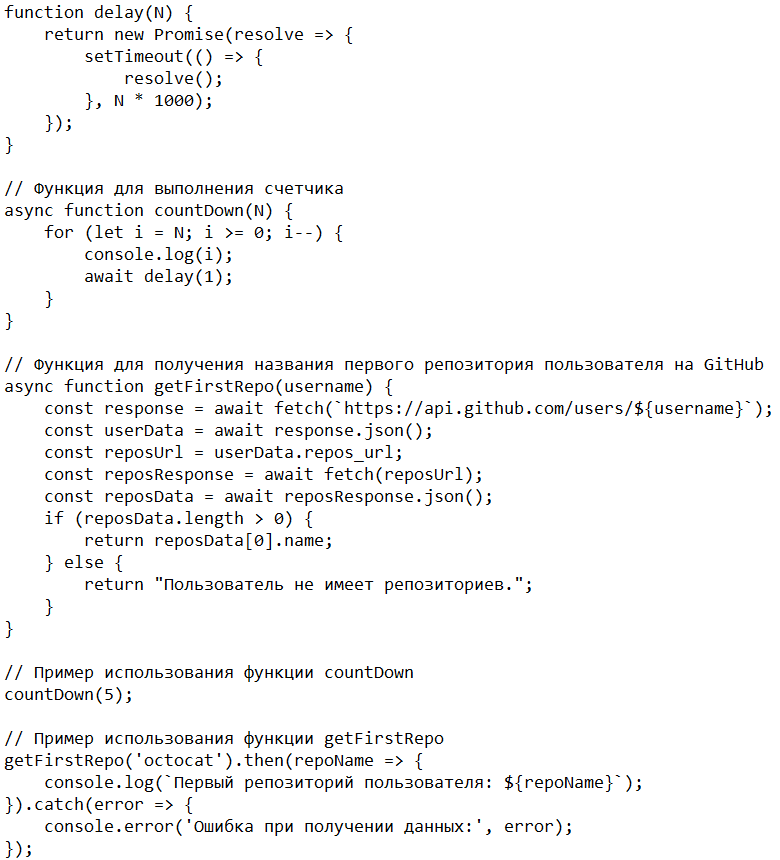
* Написать функцию counter(n), которая выводит в консоль раз в секунду числа n, n-1 ... 2, 1, 0 и останавливается.
* Написать функцию createCounter(n), возвращающую объект с методами:
  + start() -- запускает (или возобновляет) счётчик c интервалом 1 секунда: N, N-1.
  + pause() -- приостанавливает счёт, но не сбрасывает счётчик.
  + stop() -- останавливает счёт, сбрасывает счётчик.





**Задание 2 (на промисы)**

* Написать функцию delay(N), возвращающую промис, который сделает resolve() через N секунд.
* Решить задачу со счётчиком N, N-1 ... 2, 1, 0 через функцию delay.
* Написать функцию, возвращающую название первого репозитория на github.com по имени пользователя (2 последовательных запроса: [https://api.github.com/users/%USERNAME%](https://api.github.com/users/%25USERNAME%25)).



**Задание 3 (на async/await)**

Перепишите, используя async/await вместо .then/catch.

В функции getGithubUser замените рекурсию на цикл, используя async/await.

**class HttpError extends Error {**

**constructor(response) {**

**super(`${response.status} for ${response.url}`);**

**this.name = 'HttpError';**

**this.response = response;**

**}**

**}**

**function loadJson(url) {**

**return fetch(url)**

**.then(response => {**

**if (response.status == 200) {**

**return response.json();**

**} else {**

**throw new HttpError(response);**

**}**

**})**

**}**

**// Запрашивается логин, пока github не вернёт существующего пользователя.**

**function getGithubUser() {**

**let name = prompt("Введите логин?", "iliakan");**

**return loadJson(`https://api.github.com/users/${name}`)**

**.then(user => {**

**alert(`Полное имя: ${user.name}.`);**

**return user;**

**})**

**.catch(err => {**

**if (err instanceof HttpError && err.response.status == 404) {**

**alert("Такого пользователя не существует, пожалуйста, повторите ввод.");**

**return demoGithubUser();**

**} else {**

**throw err;**

**}**

**});**

**}**

**getGithubUser();**

