

# Задание на использование заглушек в модульных тестах

---

Задание выполняется в ветке `lab3`

## Задание

- Реализовать функцию `csvToJson(input: string[], delimiter: string) : object[]`, которая принимает массив строк — первая строка шапка таблицы (названия столбцов, разделённых символами, переданными через `delimiter`), остальные строки являются значениями, разделёнными символами из `delimiter`. Функция должна возвращать массив объектов.  
Например:

```
let res = csvToJson(["p1;p2;p3:p4", "1;A;b;c", "2;B;v;d"], ';');
```

Запишет в `res` массив:

```
[  
  {  
    p1: 1,  
    p2: 'A',  
    p3: 'b',  
    p4: 'c'  
  },  
  
  {  
    p1: 2,  
    p2: B,  
    p3: 'v',  
    p4: 'd'  
  }  
]
```

При реализации функции считать, что порядок столбцов и порядок следования свойств объекта совпадают, т.е., если в таблице столбцы идут в порядке `p1;p2;p3`, то и в объекте на выходе они идут также. Если есть несовпадение по типу и количеству параметров сгенерировать `Error`.

- Реализовать функцию `formatCSVFileToJsonFile(input: string, output: string, delimiter: string): Promise<void>`, которая принимает пути к файлу с таблицей в csv формате, файлу куда нужно записать таблицу переформатированную в JSON, строку с символами разделителями. Функция должна считывать данные при помощи функции `readFile` из пакета `node:fs/promises`, преобразовывать данные при помощи функции `csvToJson`, записывать результат в файл `output` при помощи функции `writeFile` из этого же пакета.

3. Реализовать модульные тесты для функции `csvToJson`. Протестировать работу на корректных входных данных и как себя ведёт функция на некорректном вводе.
4. Реализовать модульные тесты для функции `formatCSVFileToJsonFile`. При написании тестов сделать подмену функций `readFile` и `writeFile` заглушками. Для `readFile` задать, что функция возвращает, протестировать, что при указанном выходе `readFile`, `writeFile` вызывается с нужными параметрами.