Исходное тестовое задание: <https://docs.google.com/document/d/1d0pY2Bq3bOCmaEAoEduDhQdOlorw9IdM2XkQkZraFOs/edit>

### Задание 1:

Решение: <https://www.db-fiddle.com/f/uKH5HLSqDBYZmYjd1QBVVQ/5>

### Задание 2:

**Какие проблемы имеет данные подход?**

file\_put\_contents - идентична последовательным успешным вызовам функций fopen(), fwrite() и fclose() и есть вероятность того что между вызовами fopen() и fwrite() вклинится еще file\_put\_contents.

Если простыми словами, представим что 2 процесса, которые запустились в одно время. Они получили одинаковое значение счетчика, один из процессов сохранил новый счетчик в файл, а другой процесс не учел новое изменение.

Помимо этого есть еще ряд проблем:

- Файл может быть удален

- Могут быть изменены права на доступ к файлу, и файл нельзя будет изменить или прочитать

- В файл можно сохранить строку, что поломает счетчик

**Как вы их можете решить? (Нельзя использовать другие технологии)**

- Включить блокировку файла на запись LOCK\_EX. В функции file\_put\_contents выставить flags LOCK\_EX. То есть сделать так:

file\_put\_contents("./counter.txt", file\_get\_contents("./counter.txt") + 1, LOCK\_EX);

- Проверка на существование файла

- Проверка что файл доступен на чтение и запись

- Валидация данных перед сохранением, после прочтения и привести к числу

**Через некоторое время нагрузка на сервер значительно выросла. Какие проблемы вы видите?**

Из-за блокировки процессы будут ждать когда файл освободится.

**Как вы их можете решить?**

Так как мы не ожидаем ответа от счетчика, то можно сделать запуск счетчика асинхронно.

**Если бы вы могли выбрать другую технологию, то какую и почему?**

Как вариант рассмотрел бы Redis, почему:

- Есть счетчик из коробки, https://redis.io/commands/incr

- Атомарный, не нужны блокировки

- Быстрый, так как работае в памяти

- Умеет сохранять данные на диск

минус, в том, что есть вероятность потерять данные

### Задание 3:

Code review: [https://github.com/zualex/test-skyeng/pull/1](https://github.com/zualex/skyeng/pull/1)

Моя версия: [https://github.com/zualex/test-skyeng/tree/master/task3/myVersion/src/Integration](https://github.com/zualex/skyeng/tree/master/task3/myVersion/src/Integration)  
Тесты: [https://github.com/zualex/test-skyeng/tree/master/task3/myVersion/tests](https://github.com/zualex/skyeng/tree/master/task3/myVersion/tests)

### Задание 4:

Код функции: [https://github.com/zualex/test-skyeng/blob/master/task4/src/helpers.php](https://github.com/zualex/skyeng/blob/master/task4/src/helpers.php)

Тесты: [https://github.com/zualex/test-skyeng/blob/master/task4/tests/HelpersTest.php](https://github.com/zualex/skyeng/blob/master/task4/tests/HelpersTest.php)

### Задание 5:

Решение: <https://www.db-fiddle.com/f/2EJgG4AuMpwghvYFccNKRs/2>