Тестовое задание для Junior PHP Backend разработчика

Предположим, у нас есть доступ к последовательности целых чисел произвольной длины. Вся последовательность заранее неизвестна. Мы можем получать только одно число за раз (например, если читаем данные из сокета или файла). Программно мы можем оформить это в виде генератора случайных чисел:

$digits = **function**(int $length)

{

**while** ($length > 0) {

**yield** *mt\_rand*();

--$length;

}

};

Требуется:

1. Написать класс принимающий на вход последовательность произвольной длины **n** и возвращающий **m** самых больших чисел последовательности. Пример использования класса может быть таким:  
   $sequence = **new** Sequence($m);

**foreach** ($digits($n) **as** $number) {

$sequence->add($number);

}

*print\_r*($sequence->getMaxNumbers());

Или таким:  
$sequence = **new** Sequence($digits($n), $m);

*print\_r*($sequence->getMaxNumbers());

Ну или ваш вариант.

1. Добавьте в класс Sequence логирование всех основных шагов вашего алгоритма.

Код следует написать на PHP версии 7.3.\* или выше.

Как будет оцениваться тестовое задание:

1. Корректность решения. Программа должно делать ровно то, что требуется в задании.
2. Оптимальность. Чем меньше программа потребляет памяти и чем быстрее она работает, тем лучше.
3. Чистота кода. Код будет проверяться на соответствие принципам [SOLID](https://ru.wikipedia.org/wiki/SOLID_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и принципам написания “чистого” кода (т.е. насколько код читабелен и насколько легко в него можно вносить изменения).

Приемлемое время выполнения тестового задания: **не более 8 часов**.