Задание №5:

Создать класс Stack.
В качестве единственного необязательного параметра конструктор Stack должен принимать максимальное количество элементов в стеке. Если параметр является невалидным числом, генерировать ошибку. Если параметр не указан, задавать максимальный размер стека равным 10.

Реализовать публичные методы:
- push(elem) - добавить новый элемент в стек (генерировать ошибку, если стек переполнен);
- pop() - удалить верхний элемент стека и вернуть его (генерировать ошибку, если стек пуст);
- peek() - получить верхний элемент стека (вернуть null, если стек пуст);
- isEmpty() - возвращает логическое значение (пуст стек или нет);
- toArray() - возвращает новый массив, состоящий из элементов стека.

Реализовать статические публичные методы:
- fromIterable(iterable) - возвращает новый Stack, элементами которого служат элементы переданной итерируемой сущности. Максимальное количество элементов такого стека должно быть равно длине этой сущности. Если сущность не является итерируемой генерировать ошибку.

Этого достаточно для получения максимального балла по данному ДЗ.
На что буду обращать внимание:
1) В первую и главную очередь буду смотреть на понимание структуры данных. Т.е. если увижу в классе Stack что-то по типу обычного перебора элементов по индексам или же использования встроенных методов массива, то минус задание. Реализация класса должна полностью соответствовать сущности того, что вы реализуете. Если стек работает только через верхний элемент, то и "крутиться" нужно от этого.
2) Кроме этого, естественно, все функции класса должны отрабатывать корректно. Будут проверяться тестами. Однако ещё раз обращаю ваше внимание, что даже при закрытии 100% тестов можно получить очень низкий балл из-за некорректной реализации.

В репозитории от вас жду один файл с названием stack.js, в котором лежит только реализация вашего класса. Класс, повторюсь, называем Stack. Также, пожалуйста, добавьте в конце файла такую вот операцию:
module.exports = { Stack };
Это нужно, чтобы мне самостоятельно не дописывать её в каждом вашем файле.

Кому будет скучно/недостаточно, можете сделать ещё один класс LinkedList.
Конструктор этого класса не должен принимать никаких параметров.

Реализовать публичные методы:
- append(elem) - добавить элемент в конец связного списка;
- prepend(elem) - добавить элемент в начало связного списка;

- find(elem) - осуществить поиск по элементам по заданному значению. Вернуть найденный элемент или null, если такого элемента нет;
- toArray() - возвращает новый массив, состоящий из элементов связного списка.

Реализовать статические публичные методы:
- fromIterable(iterable) - возвращает новый LinkedList, элементами которого служат элементы переданной итерируемой сущности. Если сущность не является итерируемой генерировать ошибку.