Тестовое задание включает в себя задачу по разработке функционала библиотеки для работы с геометрическими фигурами.  
   
Архитектура библиотеки должна примерно повторять архитектуру общепринятых в js-разработке стандартных библиотек - <https://developer.mozilla.org/ru/docs/Web/API> . Имеется в виду соблюдение правил наименования классов (использование camelCase), переменных, функций; использование Promise, async/await; использование EventTarget для вызова событий и т.д.  
   
Библиотека должна быть реализована на языке TypeScript и компилироваться без ошибок с флагами "allowJs" : false и "strict" : true . Использование типов any и unknown допустимо в крайних случаях.

Библиотека должна предоставлять возможность создания и работы с тремя видами геометрических фигур: прямоугольник, треугольник и круг.

Под созданием фигур подразумевается получение некой структуры (объекта), позволяющей явным образом идентифицировать фигуру. Под работой с фигурой подразумевается возможность получить параметры фигуры или каким-либо образом взаимодействовать с ней.

Например, для объекта прямоугольника, параметрами могут быть ширина и высота. Под взаимодействием с прямоугольником, можно понимать такие задачи, как расчёт площади прямоугольника. Для круга - параметром будет радиус, и расчёт диаметра в качестве взаимодействия.

Набор параметров или способов взаимодействия для каждого типа фигур не специфицирован, и разрабатывается на усмотрение разработчика. Представление фигур в программном коде не специфицировано и реализуется с учетом видения правильной архитектуры у разработчика.  
  
Архитектура библиотеки должна позволять добавлять реализацию новых фигур в код библиотеки (например, трапецию или многоугольник с неограниченным количеством вершин) с минимальными временными затратами и с наиболее логически правильной организацией программного кода. Понятие, что является "наиболее логически правильным", решается разработчиком библиотеки.  
   
Плюсом будет комментирование основных методов API с синтаксисом tsdoc.

Компиляция в бандл или какие-либо дополнительные действия (обфускация, написание конфигурации webpack и т.п.) с программным кодом не требуются.

С библиотекой должен поставляться файл Readme.md, содержащий описание способов подключения и работы с библиотекой.

Исходный код желательно размещать на github.com или подобных сервисах.