**Задания по ООП JS**

1. **Класс Circle**

* Сделайте класс Circle, который будет описывать круг.
* Передайте в параметр конструктора радиус круга. Запишите его в приватное свойство.
* Сделайте метод, который будет возвращать площадь круга.
* Сделайте метод, который будет возвращать длину окружности.

1. **Класс Rectangle**

* Сделайте класс Rectangle, в котором в свойствах будут записаны ширина и высота прямоугольника.
* В классе Rectangle сделайте метод getSquare, который будет возвращать площадь этого прямоугольника.
* В классе Rectangle сделайте метод getPerimeter, который будет возвращать периметр этого прямоугольника.
* В классе Rectangle сделайте метод getRatio, который будет возращать площадь, деленную на периметр.

1. **Класс Text**

* Сделайте класс Text, который будет выполнять некоторые манипуляции над текстом. Пусть текст передается параметром конструктора.
* Сделайте метод, который вернет количество символов в тексте.
* Сделайте метод, который вернет количество букв в тексте.
* Сделайте метод, который вернет количество пробелов в тексте.
* Сделайте метод, который вернет количество символов в тексте за вычетом пробелов.
* Сделайте метод, который вернет количество символов в тексте за вычетом пробелов.
* Сделайте метод, который вернет массив предложений.

1. **Класс Period**

* Сделайте класс Period, который будет находить разницу между двумя моментами времени. Пусть эти моменты передаются параметрами конструктора.
* Сделайте метод, который вернет разницу между моментами в секундах.
* Сделайте метод, который вернет разницу между моментами в минутах.
* Сделайте метод, который вернет разницу между моментами в часах.
* Сделайте метод, который вернет разницу между моментами в днях.

1. **Класс Zate**

* Сделайте класс Zate, который будет представлять собой дату. Пусть конструктор этого класса параметрами принимает год, месяц и день даты.
* Сделайте метод, который вернет год.
* Сделайте метод, который вернет номер месяца.
* Сделайте метод, который вернет день месяца.
* Сделайте метод, который вернет номер дня недели.
* Сделайте метод, который вернет название дня недели.
* Сделайте метод, который вернет название месяца.

1. **Класс ZateExt**

* Сделайте класс ZateExt, который будет наследовать от класса Zate  и расширять его некоторыми методами.
* Сделайте метод, который будет прибавлять к дате заданное количество лет.
* Сделайте метод, который будет отнимать от даты заданное количество лет.
* Сделайте метод, который будет прибавлять к дате заданное количество месяцев.
* Сделайте метод, который будет отнимать от даты заданное количество месяцев.
* Сделайте метод, который будет прибавлять к дате заданное количество дней.
* Сделайте метод, который будет отнимать от даты заданное количество дней.

1. **Класс Month**

* Сделайте класс Month, который будет представлять собой месяц. Пусть конструктор этого класса параметром принимает номер месяца.
* Сделайте метод, который вернет номер месяца.
* Сделайте метод, который вернет название месяца.
* Сделайте метод, который вернет номер последнего дня месяца.
* Сделайте метод, который вернет номер дня недели первого дня месяца.
* Сделайте метод, который вернет номер дня недели последнего дня месяца.

1. **Класс Store**

Класс Store, представляет собой оболочку над локальным хранилищем и позволит легко сохранять и изменять в нем хранящиеся структуры данных.

Давайте посмотрим, как мы будем работать с описанным классом. Для начала создадим его объект:

let store = new Store;

Теперь сохраним данные с заданным ключом:

store.set('key', {a: 1, b: 2, c: 3});

Получим эти данные по ключу:

let res = store.get('key');

console.log(res); // {a: 1, b: 2, c: 3}

Получим часть сохраненной структуры, указав несколько ключей через точку:

let res = store.get('key.a');

console.log(res); // 1

Получим другую часть по сложному ключу:

let res = store.get('key.b');

console.log(res); // 2

* Реализуйте описанный класс.
* Сделайте метод, который будет изменять данные по заданному ключу.
* Сделайте метод, который будет удалять данные по заданному ключу.