1. Создать класс **Human**, принимающий следующие параметры:
имя **firstName** (по умолчанию задается как **"John"**), фамилию **lastName** (по умолчанию задается как **"Smith"**), возраст **age** (по умолчанию задается как **0**), пол **gender** (по умолчанию задается как **"undefined"**).
Созданный класс должен иметь метод **fullName**, не принимающий аргументы и возвращающий полное имя (имя + фамилия), а также статический метод **greetAliens(race)**, принимающий в качестве параметра название расы инопланетян и возвращающий строку **"Welcome to the Earth *raceName*"**.

2. Дан класс **Animal**:

```
class Animal {
  constructor(name, age, legs, species, status) {
    this.name = name;
    this.age = age;
    this.legs = legs;
    this.species = species;
    this.status = status;
  }
  introduce() {
    return `Hello, my name is ${this.name} and I am ${this.age} years
old.`;
  }
}
```

Создать следующие классы, наследующие от **Animal**:

I) Класс акул **Shark**, конструктор которого принимает 3 аргумента в следующем порядке: **name**, **age**, **status.** Значение свойства **legs** должно быть равно **0** для всех акул, а значение свойства **species** должно быть равно **"shark"**.
II) Класс кошек **Cat**, конструктор которого также принимает 3 аргумента: **name**, **age**, **status**. Метод **introduce** для класса **Cat** должен быть таким же, как в **Animal**, но дополнительно иметь после **"…old."** 2 пробела и слова **"Meow meow!"**. Т.е. метод **introduce()** для **Cat должен** возвращать строку **"Hello, my name is *name* and I am *age* years old. Meow meow!"**.

III) Класс собак **Dog**, конструктор которого принимает 4 аргумента в следующем порядке: **name**, **age**, **status**, **master**. Значением **master** должно быть имя хозяина собаки. Значение свойства **legs** должно быть равно **4** для всех собак, а значение свойства **species** должно быть равно **"dog"**. Для собак должен быть свой аналог метода **introduce**, который называется **greetMaster** и возвращает строку **"Hello *master*"**.

3. Создать следующие классы:
I) **Cuboid**, конструктор которого принимает 3 параметра в следующем порядке: **length**, **width**, **height**. У класса **Cuboid** должен быть геттер **surfaceArea**, возвращающий площадь поверхности кубоида, а также геттер **volume**, возвращающий объем кубоида.

II) **Cube**, являющийся подклассом класса **Cuboid**. Конструктор класса **Cube** должен принимать только один параметр **length** и использовать его в качестве значения **length**, **width** и **height**.

4. Создать класс **Cube**, конструктор которого принимает 1 параметр – длину стороны куба **length**. Определить геттеры и сеттеры для каждого из 2 свойств: **surfaceArea** (площадь поверхности куба) и **volume** (объем куба).

5. Дан класс **Person**, для которого реализован геттер **getName**:

```
class Person {
  constructor(firstName, lastName) {
    this.firstName = firstName;
    this.lastName = lastName;
  }
  getName() {
    return this.firstName + ' ' + this.lastName;
  }
}
```

Используя метод **Object.defineProperty** добавить новое свойство **name**, которое будет являться геттером и сеттером. Геттер **name** должен возвращать строку **"*firstName* *lastName*"**, а сеттер **name** должен позволять задавать полное имя (имя + фамилия).

Пример работы:

```
let ivanPetrov = new Person("Ivan", "Petrov");
console.log(ivanPetrov.name) // "Ivan Petrov" – сработал геттер
ivanPetrov.name = "Petr Ivanov" // сработал сеттер
console.log(ivanPetrov.firstName) // "Petr"
console.log(ivanPetrov.lastName) // "Ivanov"
```