

Работа с объектами

1. Сравнение

Напишите функцию, которая будет сравнивать 2 объекта на совпадение всех ключей (свойств). Т.е. принимать 2 объекта и возвращать true, если у этих объектов все ключи одинаковые.

2. Удаление

Напишите функцию для удаления заданных свойств у объекта. Функция должна первым параметром принимать объект, а следующими параметрами – перечисление свойств, который нужно удалить. Очищенный объект должен возвращаться в конце функции.

```
let obj = { a: 2, mambo: 43, zelda: 10, lock: "333" };
obj = deleteKeys(obj, "mambo");
// obj = { a: 2, zelda: 10, lock: "333" }

obj = { a: 2, mambo: 43, zelda: 10, lock: "333" };
obj = deleteKeys(obj, "mambo", "lock");
// obj = { a: 2, zelda: 10 }

obj = { a: 2, mambo: 43, zelda: 10, lock: "333" };
obj = deleteKeys(obj, "mambo", "knight", "loop", "lock", "rest", "a", "b");
// obj = { zelda: 10 }
```

3. Сортировка

Дан следующий массив из объектов:

```
const library = [
  {
    year: 1995,
    pages: 286,
    title: "The Road Ahead",
    author: "Bill Gates",
    libraryID: 1254
  },
  {
    year: 2015,
    pages: 656,
    title: "Steve Jobs",
    author: "Walter Isaacson",
    libraryID: 4264
  },
  {
    year: 2008,
    pages: 464,
```

```

        title: "Clean Code: A Handbook of Agile Software
Craftsmanship",
        author: "Robert C. Martin",
        libraryID: 3245
    },
    {
        year: 2015,
        pages: 706,
        title: "Cracking the Coding Interview: 189 Programming
Questions and Solutions",
        author: "Gayle Laakmann McDowell",
        libraryID: 2748
    },
    {
        year: 1997,
        pages: 672,
        title: "The Art of Computer Programming, Vol. 1: Fundamental
Algorithms, 3rd Edition",
        author: "Donald E. Knuth",
        libraryID: 2748
    }
    ]];

```

Выполните различные варианты сортировок:

- по возрастанию года издания;
- по убыванию года издания;
- по возрастанию количества страниц;
- по имени автора.

4. Машинки

Сейчас мы будем создавать цех по сборке автомобилей. Нужно разработать цепочку из функций, которые будут добавлять к объекту будущего автомобиля новые свойства-детали. В каждую из этих функций должен передаваться объект, добавляться свойство, и возвращаться исходный объект с новым свойством.

1. Функция `addEngine` должна добавлять свойство `engine` с объектом, например, таким:

```

{
    model: "AMR-250-90",
    type: "AC",
    weight: 150,
    power: 210,
    torque: 280
}

```

2. Функция `addBattery` должна добавлять свойство `battery`:

```
{
  config: "40S-10P",
  cells: 400,
  energy: 24.5,
  mass: 267
}
```

3. Функция addController для добавления контроллера (свойство controller):

```
{
  model: "Evnetics Shiva",
  current: 3000,
  power: 1.275,
  voltage: "8-425 Volts",
  prechargeCircuit: true
}
```

4. Функция addCharger должна добавлять charger:

```
{
  model: "Elcon PFC5000 Charger",
  input: "0.2 to 30A AC",
  output: "16A to 80A DC",
  algorithm: "programmable",
  efficiency: 92
}
```

5. Функция addConverter для добавления converter:

```
{
  type: "DC-DC",
  inputCurrent: "0-3 Amps / 120 volts",
  outputCurrent: "0-30 Amps / 12 volts",
  isolated: "Selectabel"
}
```

6. Функция addWheels должна добавлять свойство wheels с массивом из 4 колес:

```
{
  motor: "M700",
  drumBrake: "standard",
  rim: 15,
  tire: "265/45-R20",
  size: "20/20"
}
```

7. Функция addBody для добавления корпуса автомобиля body:

```
{
    doors: 4,
    color: "red",
    glass: "laminated",
    bumper: "S3E6-7",
    interior: "black"
}
```

Теперь нужно показать вызов этих функций двумя способами.

Напишите функцию `manufactureCar_sep`, которая будет вызывать эти функции по очереди друг за другом, разными выражениями (для каждого вызова функции новая строка кода).

Напишите функцию `manufactureCar_chain`, которая будет вызывать эти функции цепочкой, в виде одного выражения (все вызовы одной строкой кода).

Оставьте какой-нибудь один метод `manufactureCar`.

Создайте объект `Factory`, и добавьте функцию `manufactureCar` в него, чтобы она стала методом.

Далее работайте с этим методом через этот объект.

5. Статистика

Добавьте в каждую функцию производства деталей разный шанс (0.2–2%), что произойдёт брак, и в объект будет встроено что-то не то.

Напишите функцию контроля качества – для проверки готового автомобиля на наличие брака. Передаваться в неё должен собранный объект автомобиля, и она должна возвращаться `true` или `false`, в зависимости от того, всё ли в машине на месте, и нет ли чего лишнего.

На основе статистического анализа вычислите итоговый шанс брака при производстве ваших автомобилей.

6. *Замыкания

Перепишите функции добавления компонентов таким образом, чтобы они больше не принимали объект автомобиля как параметр, и не возвращали его `return`-ом, а возвращали `true/false` в зависимости от создания брака.

Доступ к объекту автомобиля нужно осуществлять с помощью замыкания. Для этого перенесите объявление всех функций добавления деталей внутрь главной функции создания машины `manufactureCar`.

Измените алгоритм создания машины таким образом, чтобы в случае производства бракованной детали, все следы брака из объекта удалялись и происходило повторное добавление неудачной детали. В случае брака это должно повторяться по циклу, пока не будет произведена качественная деталь. И так для каждой детали.

В итоге должно получиться, что в итоговом производстве автомобилей брака не будет. Проверьте это статистически.