1. Создайте объект city1 (var city1 = {}), укажите у него свойства name (название города, строка) со значением «ГородN» и population (населенность города, число) со значением 10 млн.
2. Создайте объект city2 через нотацию {name: "ГородM", population: 1e6}.
3. Создайте у объектов city1 и city2 методы getName(), которые вернут соответствующие названия городов
4. Создайте методы exportStr() у каждого из объектов. Этот метод должен возвращать информацию о городе в формате «name=ГородN\npopulation=10000000\n». Для второго города будет строка со своими значениями. Примечание: можно обращаться к каждому свойству через цикл for/in, но методы объекта возвращать не нужно
5. Создайте глобальную функцию getObj(), которая возвращает this. А у каждого из объектов city1 или city2 метод getCity, который ссылается на getObj. Проверьте работу метода. Примечание: к объекту вызова можно обратиться через this.
6. (\*) Создать объект obj, с методами method1(),method2() и method3(). В методе method3() должна возвращаться строка «метод3». Сделайте так, чтобы было возможно выполнение кода obj.method1().method2().method3().
7. Создайте массив d1 с числовыми величинами 45,78,10,3. Добавьте в массив d1 еще одно число ( d1[7] = 100). Выведите в консоль весь массив и его элементы с индексами 6 и 7.
8. Создайте массив d2 с числовыми величинами 45,78,10,3. Посчитайте в переменную sum2 сумму чисел в нем, при помощи цикла for .
9. Создайте массив d3 с числовыми величинами 45,78,10,3. Добавьте в массив d3 еще одно число (например, d[7] = 100) Посчитайте в переменную sum3 сумму чисел в нем, при помощи цикла for/in.
10. Создайте массив d4 с числовыми величинами 45,78,10,3. Напишите функцию сортировки my(a,b), которая при вызове d4.sort(my) отсортирует элементы массива по убыванию чисел. Вызовите d4.sort(my)
11. (\*) Создайте двумерный массив d5 размером n=3 элементов в каждом из которых будет m=4 элементов из целых чисел (в каждый элемент поместить цифру 5). Примечание: при создании двумерного массива используйте циклы. Во внешнем цикле должна появиться конструкция d5[i] = [];
12. **Векторный тип.** Напишите конструктор Vector, представляющий вектор в двумерном пространстве. Он принимает параметры x и y (числа), которые хранятся в одноимённых свойствах.

Дайте прототипу Vector два метода, plus и minus, которые принимают другой вектор в качестве параметра, и возвращают новый вектор, который хранит в x и y сумму или разность двух (один this, второй — аргумент)

Добавьте геттер length в прототип, подсчитывающий длину вектора – расстояние от (0, 0) до (x, y).

Изображение выглядит как текст

Автоматически созданное описание

1. Создайте тип ячейки StretchCell(inner, width, height), соответствующий интерфейсу ячеек таблицы из этой главы. Он должен оборачивать другую ячейку (как делает UnderlinedCell - из примера книги), и убеждаться, что результирующая ячейка имеет как минимум заданные ширину и высоту, даже если внутренняя ячейка была бы меньше.

Изображение выглядит как текст

Автоматически созданное описание

1. Разработайте интерфейс, абстрагирующий проход по набору значений. Объект с таким интерфейсом представляет собой последовательность, а интерфейс должен давать возможность в коде проходить по последовательности, работать со значениями, которые её составляют, и как-то сигнализировать о том, что мы достигли конца последовательности.

Задав интерфейс, попробуйте сделать функцию logFive, которая принимает объект-последовательность и вызывает console.log для первых её пяти элементов – или для меньшего количества, если их меньше пяти.

Затем создайте тип объекта ArraySeq, оборачивающий массив, и позволяющий проход по массиву с использованием разработанного вами интерфейса. Создайте другой тип объекта, RangeSeq, который проходит по диапазону чисел (его конструктор должен принимать аргументы from и to).

Изображение выглядит как текст

Автоматически созданное описание

1. Создайте конструктор function Card(from, to){...}, создающий объекты карточки со свойствами from, to и методом show(), возвращающим свойства отдельного объекта в виде строки «from,to», где на месте названий свойств будут их значения. Создайте переменную c1 с городами Екатеринбург и Москва
2. Опишите класс Human: для создания объектов со свойствами name, age и height конструктор класса должен принимать одноимённые аргументы . Создайте массив humans из десяти объектов:

«Коля», 23, 180,

«Даша», 19, 170,

«Ваня», 18, 192,

«Петя», 45, 178,

«Вася», 34, 197,

«Джони», 40, 168,

«Катя», 37, 160,

«Петя», 29, 200,

«Соня», 21, 172,

«Женя», 25, 175

* 1. Укажите классу метод getInfo (он должен возвращать строки вида «Коля, 23, 180»), метод геттер firstname ( он должен возвращать свойство name объекта)

1. (\*) Напишите функции сортировки sortByName и sortByHeight, которые смогут отсортировать массив по именам (в алфавитном порядке) или росту (по убыванию роста). Вызовите только функцию сортировки по росту.
2. Получите в переменную dt1 дату соответствующую 0:00, 1 января 2045 года, при помощи конструктора Date()
3. В переменную dt2 найдите кол-во секунд, которое прошло с 1 января 1970 года по текущий момент.
4. Напишите функцию getDays, которая принимает год и месяц (пусть январь будет — 1, февраль — 2 и т.д.), а возвращает количество дней в месяце
5. (\*) Напишите функцию calendar(year,month), которая принимает два параметра — номер месяца и год, а возвращает разметку календаря HTML на указанный месяц.
6. (\*) Через прототип расширьте встроенный объект Number методом isOdd(), который возвращает true, если число нечетное.
7. (\*) Создайте конструктор , который поддерживает на своих методах цепочечный синтаксис.
8. (\*) Представьте, что разрабатываете игру. Создайте конструктор Unit(), создающий объекты с координатами x, y. При помощи прототипного наследования расширьте Unit(), создав новый конструктор Fighter() с свойством power. Примечание: не забудьте о геттерах и сеттерах.