

# Язык JavaScript (дополнительный материал)

## Задача

Сравнить скорости операций unshift и shift для массива и для двусвязного списка.

# Задача

Добавить новые методы в массивы:

- метод **remove(index)** – возвращает новый массив без удаленного элемента;
- метод **min()** – возвращает минимальный элемент массива;
- метод **average()** – возвращает среднее арифметическое значение элементов.

# Шаблон цепочек

**Шаблон цепочек** предоставляет возможность вызова методов объектов друг за другом без необходимости присваивать предыдущее возвращаемое значение промежуточной переменной и размещать инструкции вызова в нескольких строках:

```
myobj.method1("hello")  
      .method2()  
      .method3("world")  
      .method4();
```

Метод объекта, который не имеет возвращаемого значения, всегда может возвращать из него ссылку **this**.

# Шаблон цепочек

## Достоинства:

- помогает сэкономить на вводе с клавиатуры и писать более компактный программный код, который читается почти как обычное предложение;
- шаблон заставляет задумываться о разбиении программного кода на маленькие и более узкоспециализированные функции.

## Недостаток:

- такой программный код сложнее в отладке.

# Задача

Разработать объект методы которого реализуют шаблон цепочек. Объект содержит:

- свойство **value** для хранения числа;
- метод **increment()** для увеличения свойства value на 1;
- метод **add(num)** для увеличения свойства value на num;
- метод **outAlert()** для вывода свойства value через alert.

# Шаблон единственный объект

Идея шаблона заключается в том, чтобы при создании нескольких объектов с помощью оператора `new` и одного и того же конструктора вы получали бы просто новые ссылки на один и тот же объект.

```
var storage1 = new Storage();  
var storage2 = new Storage();  
storage1 === storage2 //true
```

# Задача

Создать шаблон объектов Storage реализующих шаблон проектирования «Единственный объект» двумя способами:

- сохранить ссылку на экземпляр в статическом свойстве конструктора;
- сохранить экземпляр в замыкании.



## Шаблон мемоизации

Идея шаблона заключается в кэшировании результатов (возвращаемого значения) функции, чтобы при следующем обращении к функции ей не приходилось выполнять сложные и продолжительные вычисления.

# Шаблон мемоизации

```
function calculation(x, y) {  
    if (!calculation.memento)  
        calculation.memento = {};  
  
    var key = x.toString() + ":" + y.toString();  
    var result = 0;  
  
    if (!calculation.memento[key])  
    {  
        for (var i = 0; i < y; ++i) result += x;  
        calculation.memento[key] = result;  
    }  
    return calculation.memento[key];  
}
```

# Задача

Используя шаблон мемоизации улучшите следующий код:

```
function fibRec(n){  
    if( n === 1 || n === 2 ) {  
        return 1;  
    } else {  
        return fibRec(n-1) + fibRec(n-2);  
    }  
}
```