

Язык JavaScript (дополнительный материал)

Управлением памятью

Главной концепцией управления памятью в JavaScript является принцип достижимости (англ. reachability).

Определённое множество значений считается достижимым изначально, в частности:

- **Значения, ссылки на которые содержатся в стеке вызова**, то есть – все локальные переменные и параметры функций, которые в настоящий момент выполняются или находятся в ожидании окончания вложенного вызова.

- **Все глобальные переменные.**

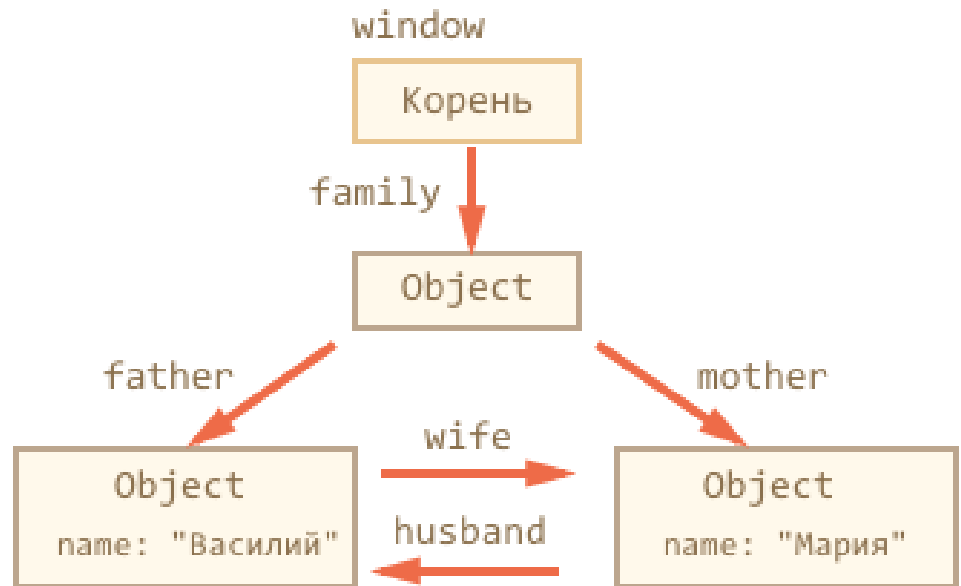
Эти значения гарантированно хранятся в памяти. Мы будем называть их корнями.

- **Любое другое** значение сохраняется в памяти лишь до тех пор, **пока доступно из корня по ссылке или цепочке ссылок.**

Управлением памятью

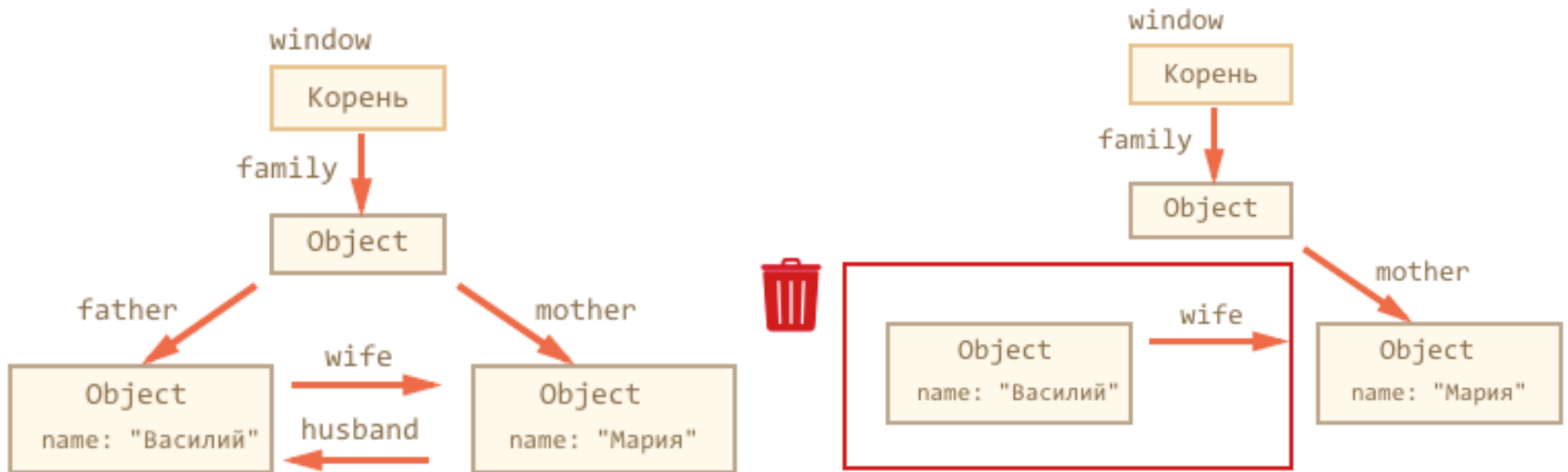
Для очистки памяти от недостижимых значений в браузерах используется автоматический Сборщик мусора.

```
1 function marry(man, woman) {  
2   woman.husband = man;  
3   man.wife = woman;  
4  
5   return {  
6     father: man,  
7     mother: woman  
8   }  
9 }  
10  
11 var family = marry({  
12   name: "Василий"  
13 }, {  
14   name: "Мария"  
15 });
```



Управлением памятью

```
1 delete family.father;  
2 delete family.mother.husband;
```



Односвязный список

Односвязный список – структура данных, в которой каждый элемент (узел) хранит информацию, а также ссылку на следующий элемент. Последний элемент списка ссылается на NULL.



Односвязный список

Достоинства:

- эффективное (за константное время) добавление и удаление элементов;
- размер ограничен только объёмом памяти компьютера и разрядностью указателей;
- динамическое добавление и удаление элементов.

Недостатки:

- сложность прямого доступа к элементу, а именно определения элемента по его индексу (порядковому номеру) в списке;
- некоторые операции со списками медленнее, чем с массивами, так как к произвольному элементу списка можно обратиться, только пройдя все предшествующие ему элементы;
- на поля-указатели (указатели на следующий элемент) расходуется дополнительная память.

Задача: Реализовать односвязный список

Можно выделить две основные части односвязного списка:

1. Шаблон объектов «Node»:

свойство **data** - здесь хранятся значения;

свойство **next** - указывает на следующий узел в списке.

2. Шаблон объектов «SingleLinkedList»:s

свойство **length** - извлекает количество узлов в списке;

свойство **head** - определяет узел, как головной элемент списка;

метод **push(value)** - добавляет в «голову» списка узел;

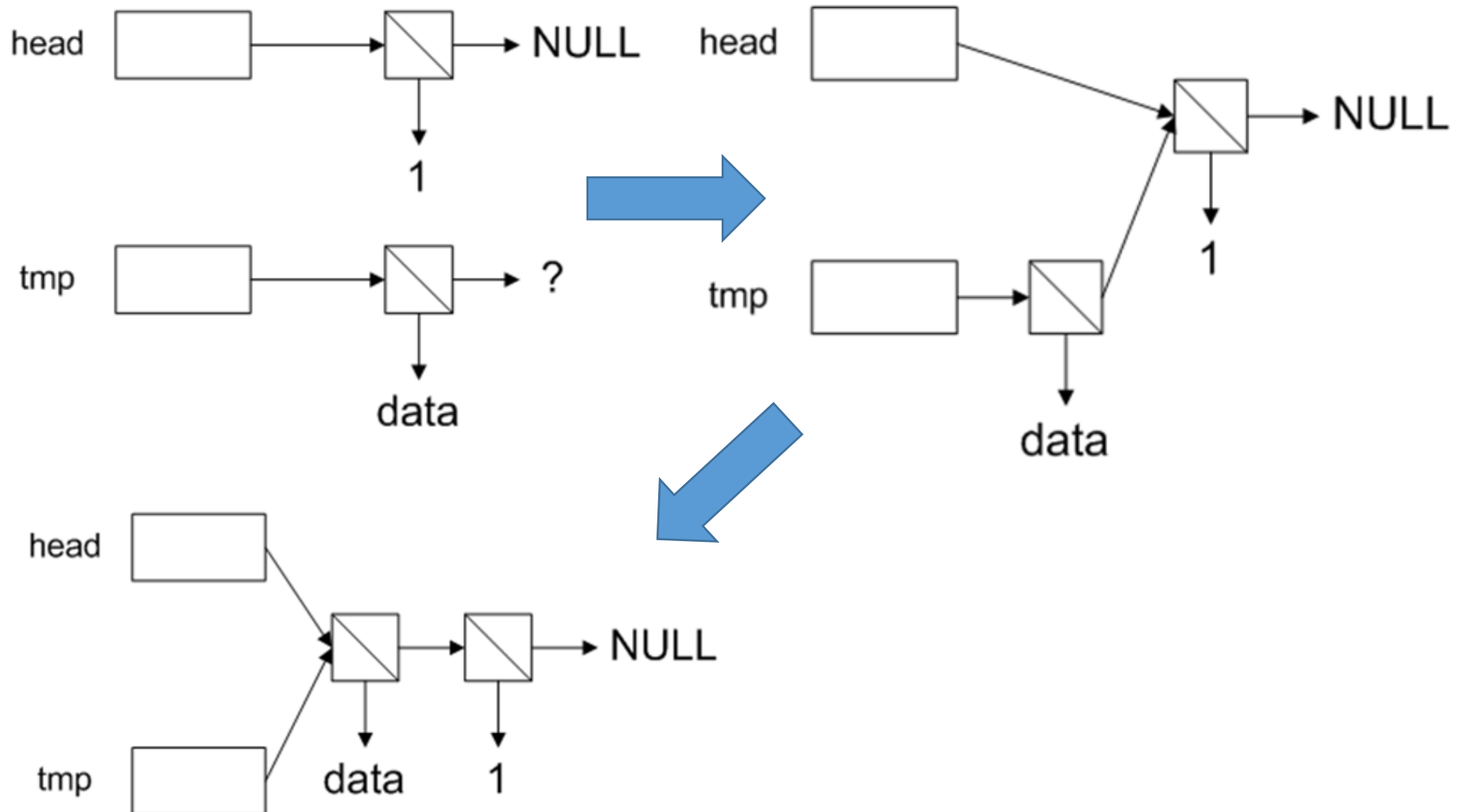
метод **pop()** - удаляет узел из «головы» списка и возвращает его содержимое;

метод **toString()** – преобразование к строке содержимого списка;

метод **unshift(value)** - добавляет в «хвост» списка узел.

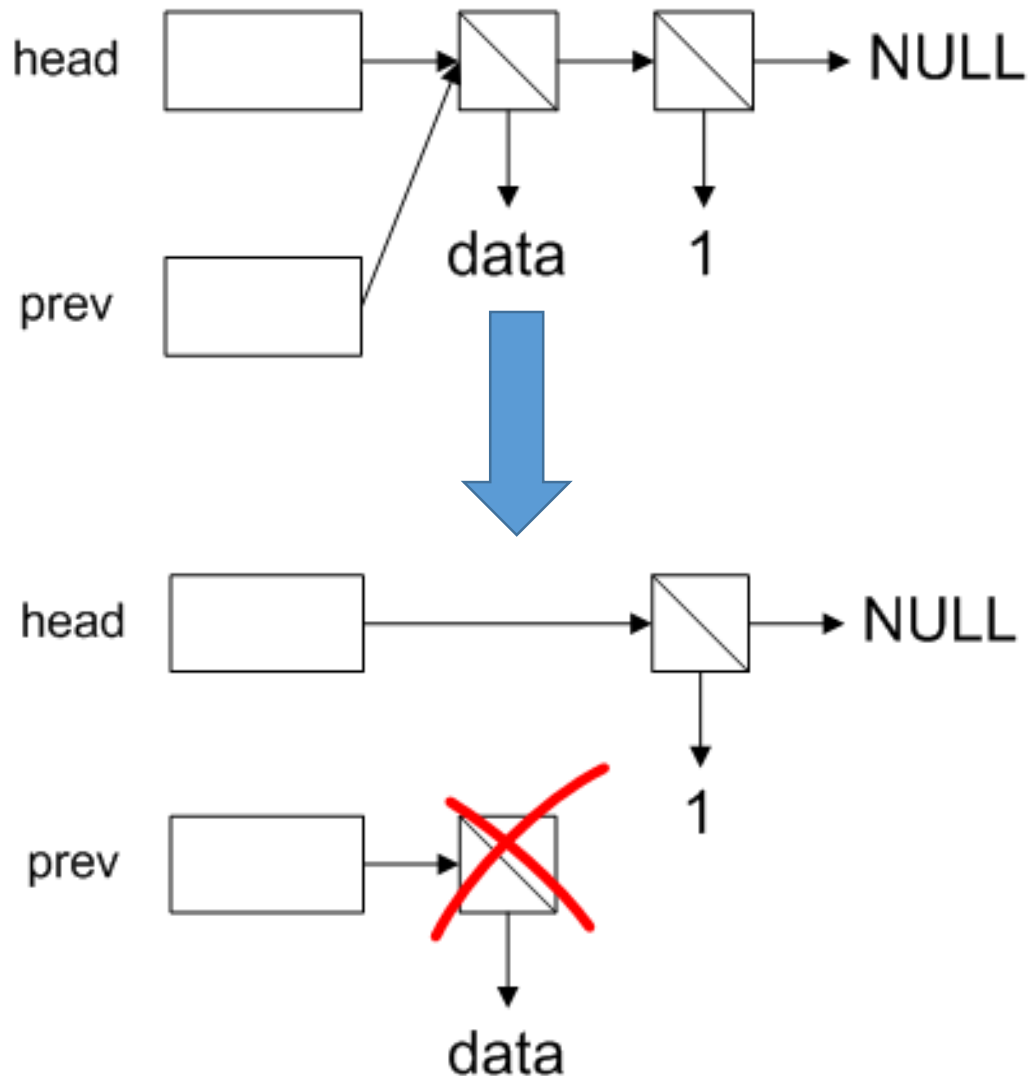
Задача: Реализовать метод push(value)

Алгоритм работы метода **push(value)**:



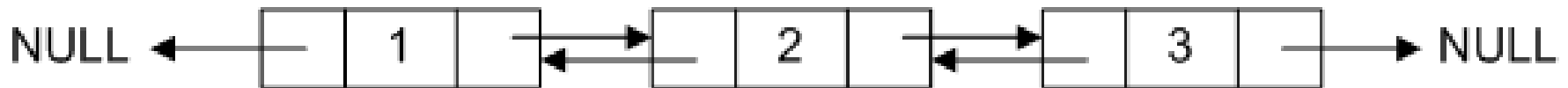
Задача: Реализовать метод pop()

Алгоритм работы метода **pop()**:



Двусвязный список

Двусвязный список - это структура данных, которая состоит из узлов, которые хранят полезные данные, указатели на предыдущий узел и следующий узел.



Задача: Реализовать двусвязный список

Можно выделить две основные части двусвязного списка:

1. Шаблон объектов «Node»:

свойство **data** - здесь хранятся значения;

свойство **next** - указывает на следующий узел в списке;

свойство **prev** - указывает на предыдущий узел в списке;

2. Шаблон объектов «DoubleLinkedList»:s

свойство **length** - извлекает количество узлов в списке;

свойство **head** - определяет узел, как головной элемент списка;

свойство **tail** - определяет узел, как хвостовой элемент списка;

метод **push(value)** - добавляет в «голову» списка узел;

метод **pop()** - удаляет узел из «головы» списка и возвращает его содержимое (данные);

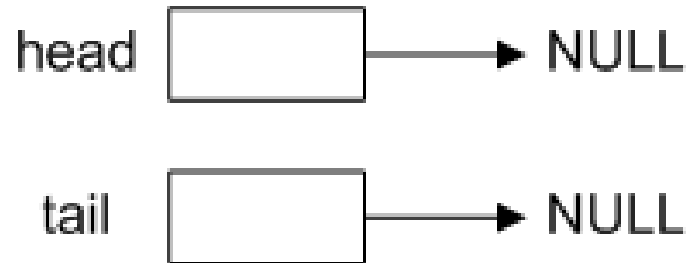
метод **toString()** – преобразование к строке содержимого списка;

метод **unshift(value)** - добавляет в «хвост» списка узел.

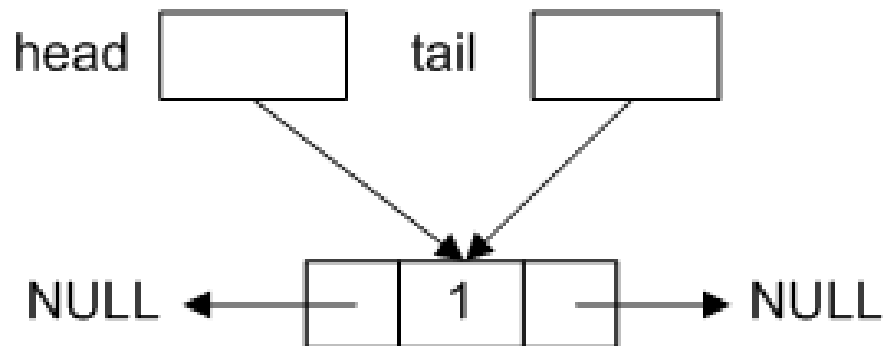
метод **shift(value)** – удаляет узел из «хвоста» списка и возвращает его содержимое (данные).

Задача: Реализовать двусвязный список

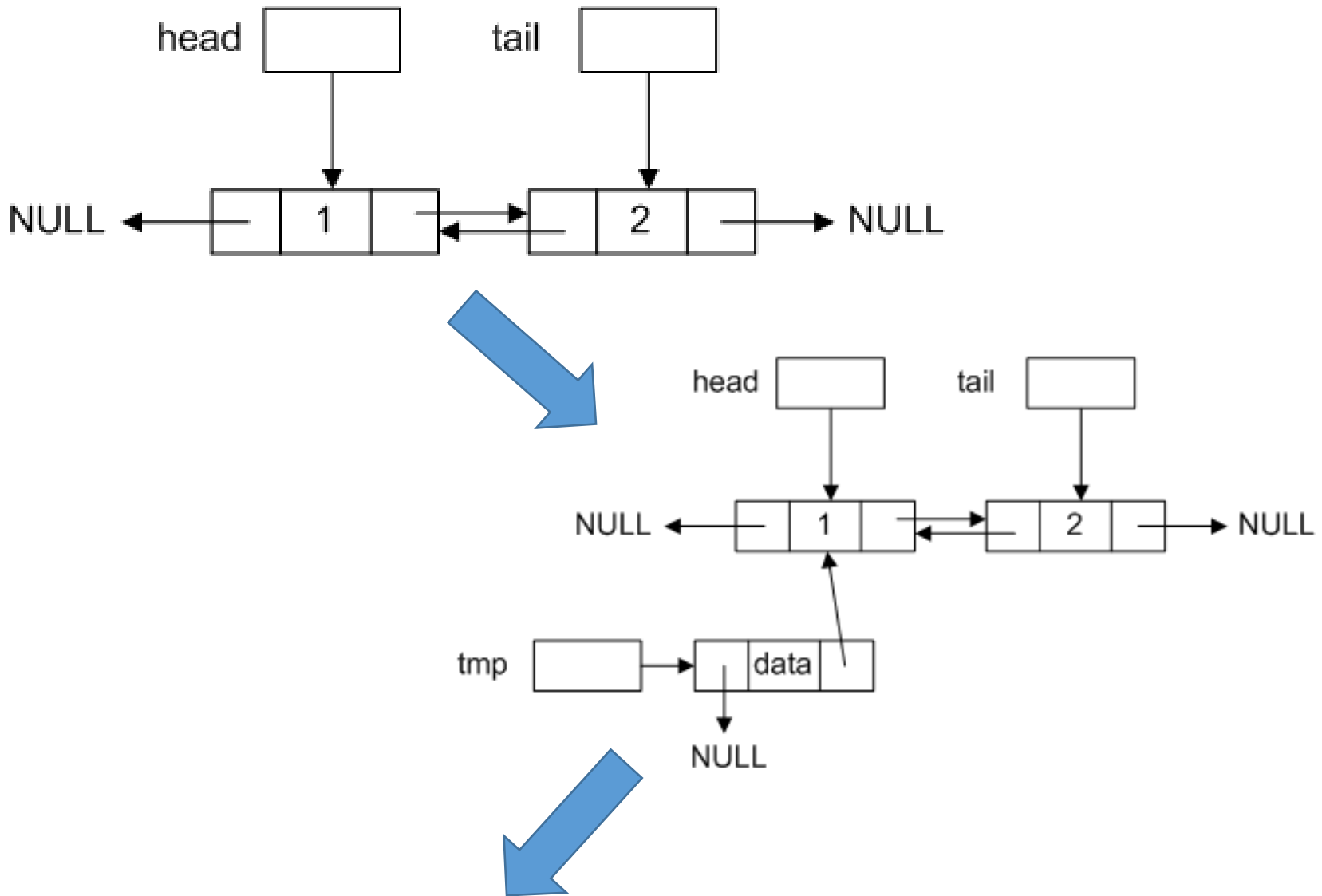
В случае, когда в списке нет элементов, оба они равны нулю.



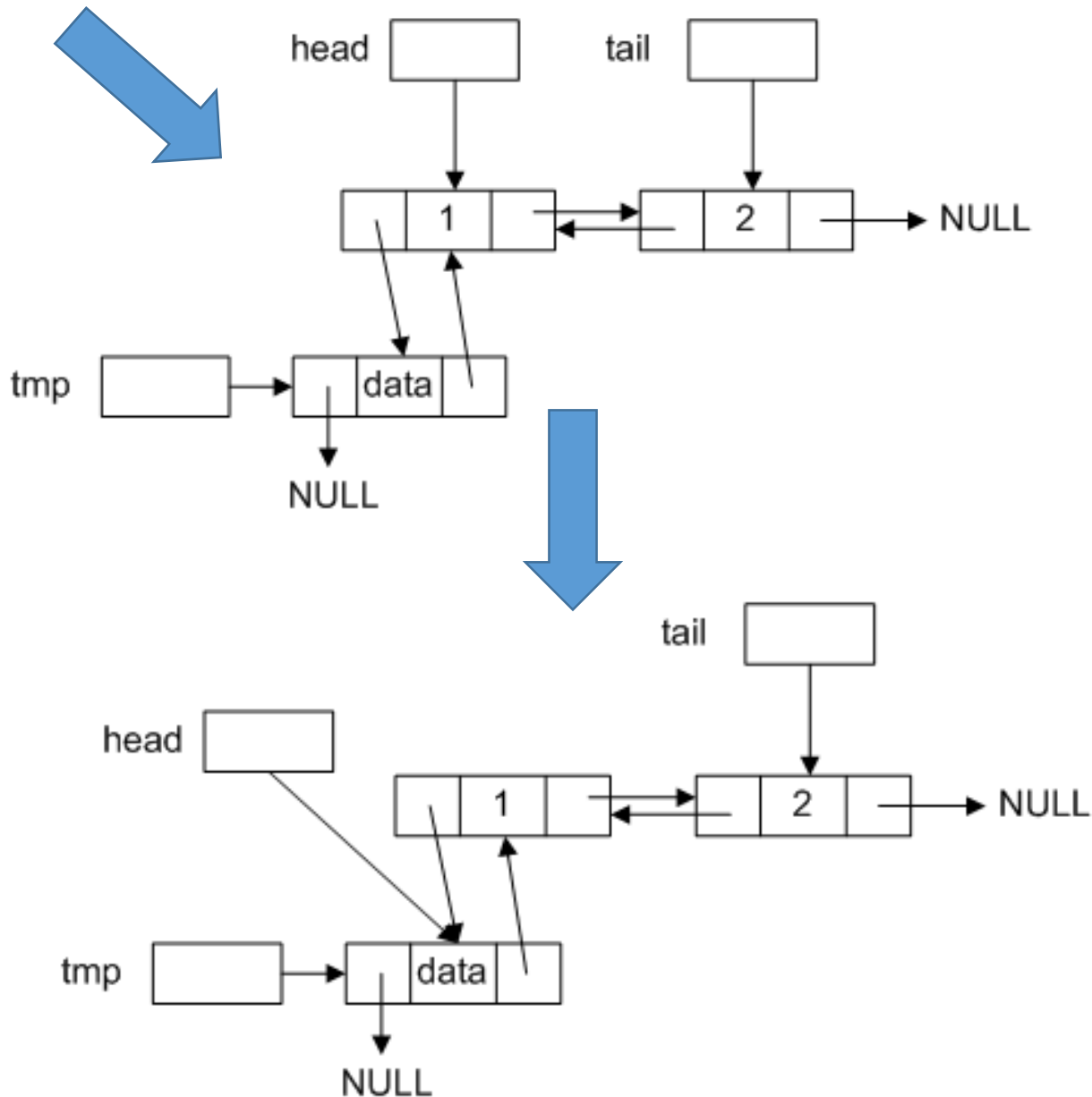
Если в списке один элемент, то оба указателя ссылаются на один и тот же элемент (соответственное, они равны).



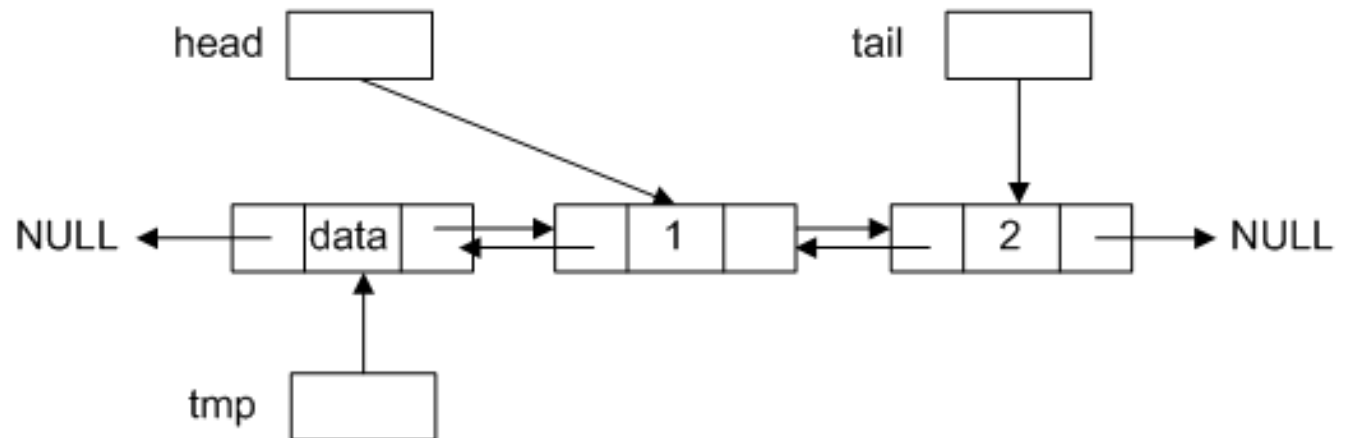
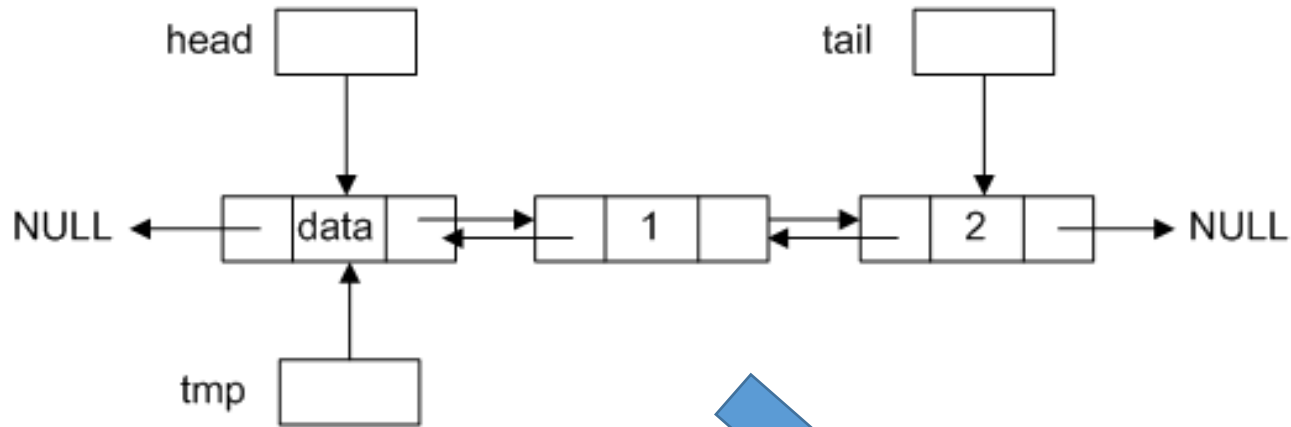
Задача: Реализовать метод push(value)



Задача: Реализовать метод push(value)



Задача: Реализовать метод pop()



Задача: Реализовать метод pop()

