

Java Professional module #2

List implementations. LinkedList and ArrayList.

Mentor: Rustam Khakov

Lists implementations. LinkedList.

- Что такое LinkedList
- Операции над LinkedList
 - Создание связанного списка
 - Добавление объекта в конец связанного списка
 - Добавление объекта в середину связанного списка
 - Удаление объекта из списка
 - Clear()
 - Contains()
- LinkedList VS ArrayList

LinkedList

LinkedList — класс, реализующий два интерфейса — ***List*** и ***Deque***.

- Это обеспечивает возможность создания двунаправленной очереди из любых (в том числе и *null*) элементов.
- Каждый объект, помещенный в связанный список, является узлом (нодом).
- Каждый узел содержит элемент, ссылку на предыдущий и следующий узел.
- Фактически связанный список состоит из последовательности узлов, каждый из которых предназначен для хранения объекта определенного типа.

Строение LinkedList

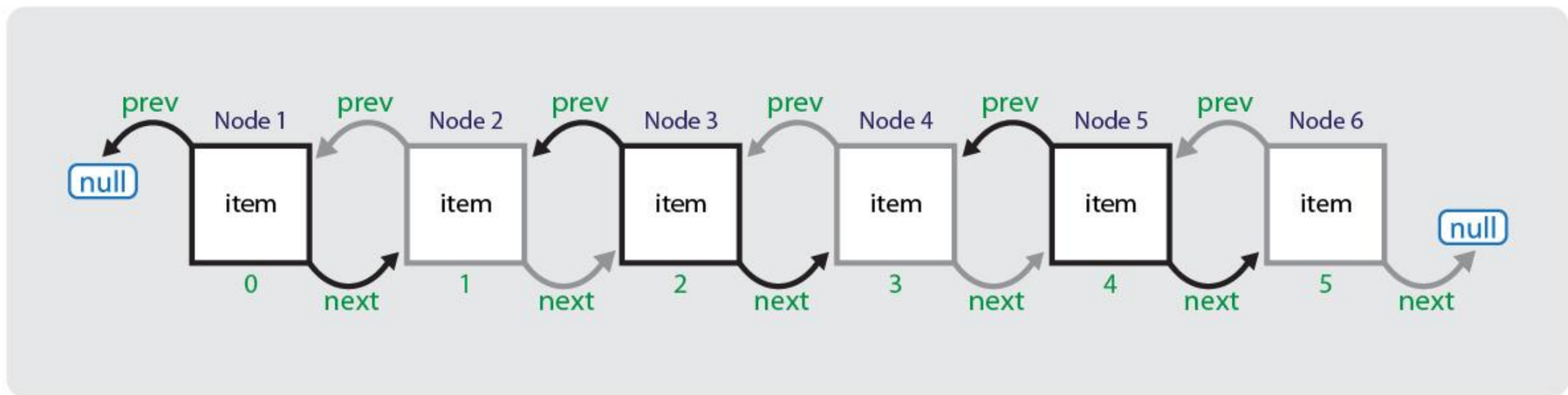


Рис. 1: Общий вид связанного списка

Создание связанного списка

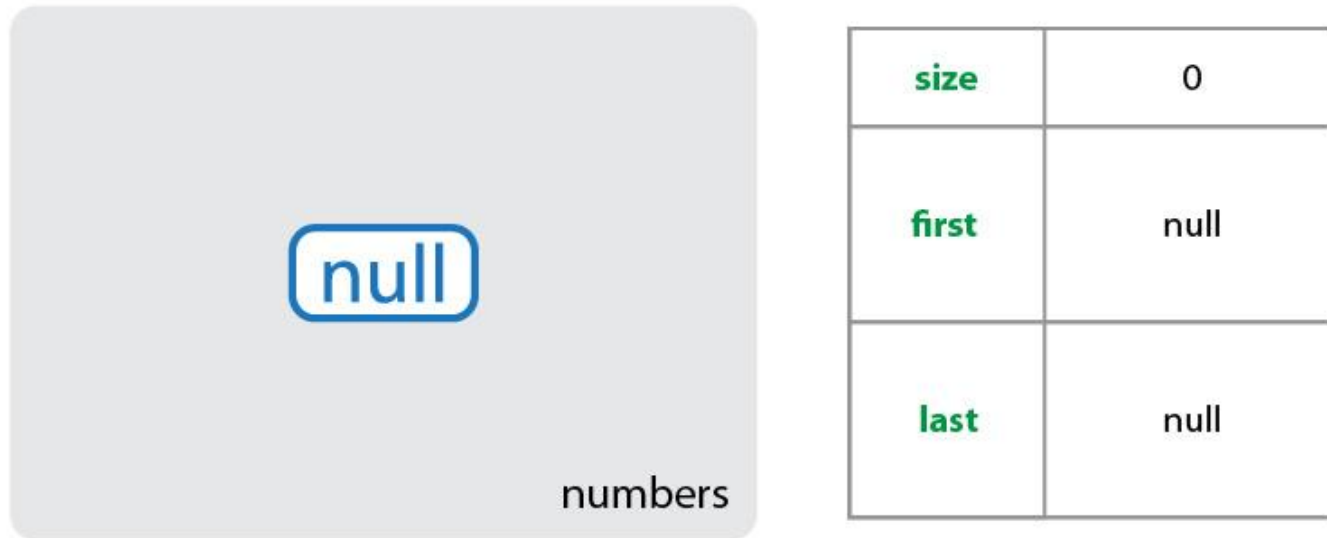


Рис. 2: Состояние объекта сразу после создания

Добавление объекта в конец связанного списка

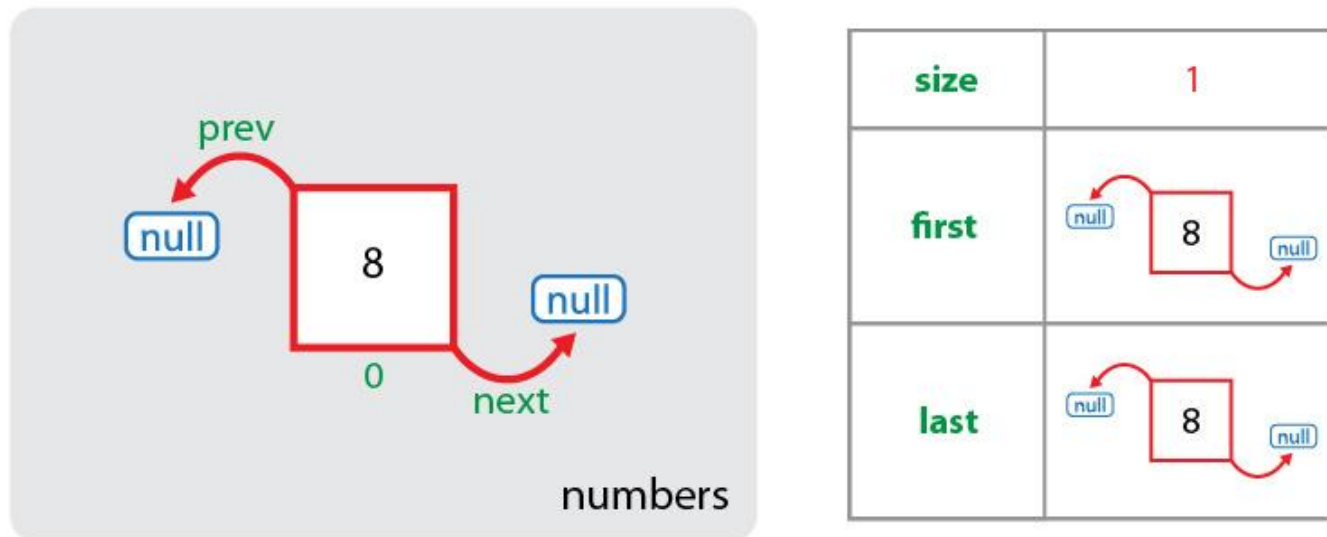
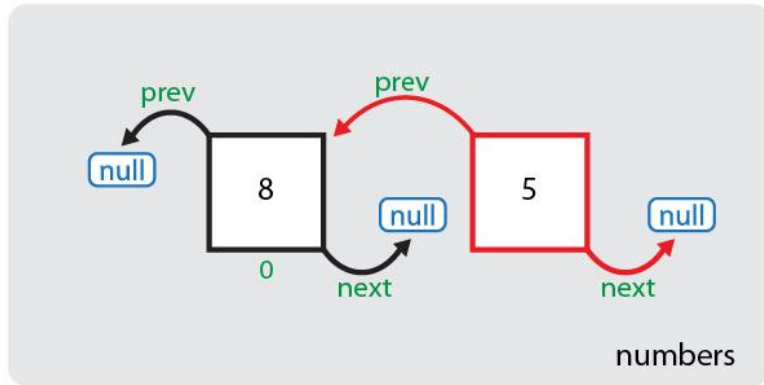


Рис. 3: Добавление первого объекта в связанный список

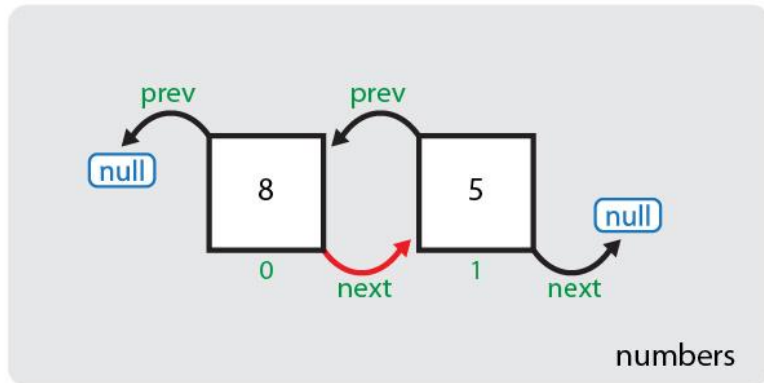


size	1
first	
last	

Рис. 4: Добавление второго объекта в связанный список (этап 1)

Добавим еще один элемент в нашу коллекцию:

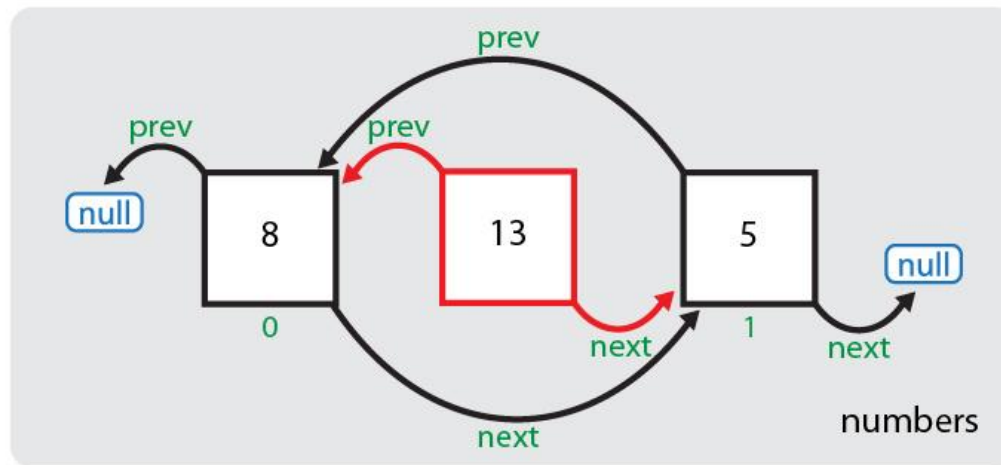
```
numbers.add(5)
;
```



size	2
first	
last	

Рис. 5: Добавление второго объекта в связанный список (этап 2)

Добавление объекта в середину связанного списка



size	2
first	
last	

Рис. 6: Добавление объекта в середину связанного списка (этап 1)

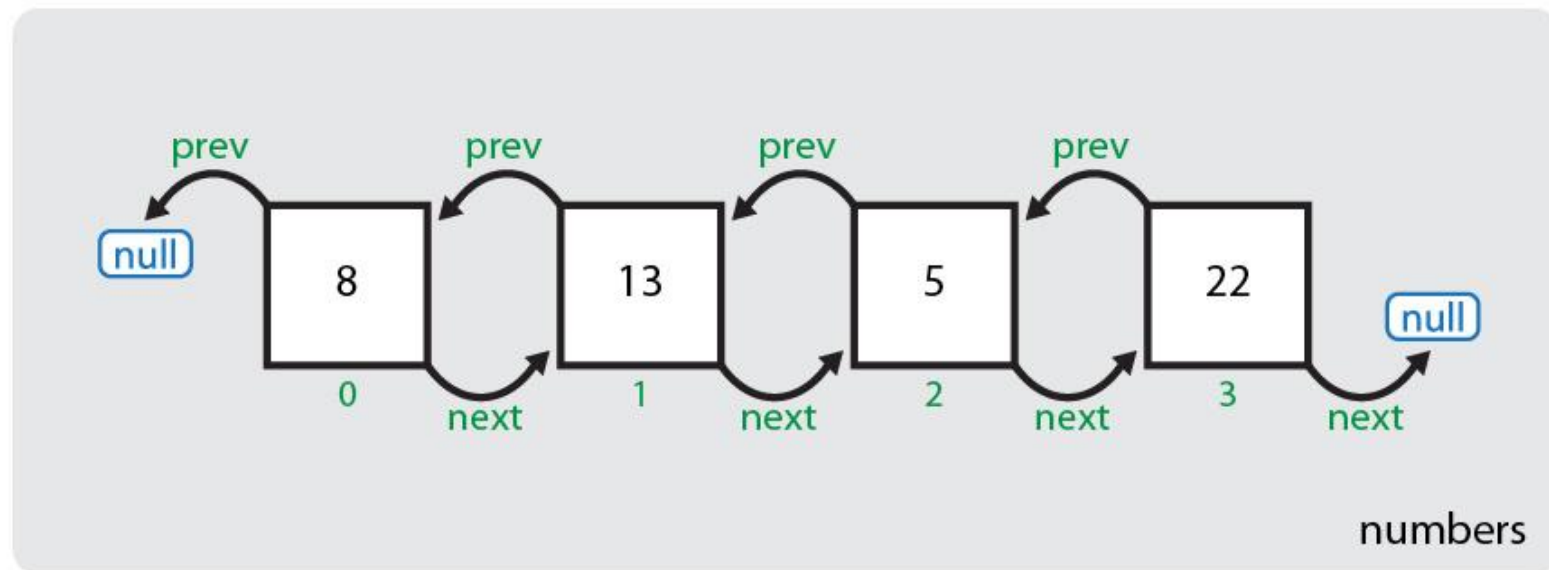
Удаление объекта из списка

Метод	Удаляемый элемент	Если найден	Если не найден
poll()	первый	возвращает удаленный элемент	возвращает null
pollFirst()			
remove()			NoSuchElementException
removeFirst()			
pollLast()	последний		возвращает null
removeLast()			NoSuchElementException
remove(index)	элемент в указанной позиции		
remove(object)	первое вхождение указанного объекта	возвращает true	возвращает false
removeFirstOccurrence(object)			
removeLastOccurrence(object)	последнее вхождение указанного объекта		

Табл. 1: Методы LinkedList для удаления элемента

Рассмотрим удаление элемента из связанного списка по его значению.

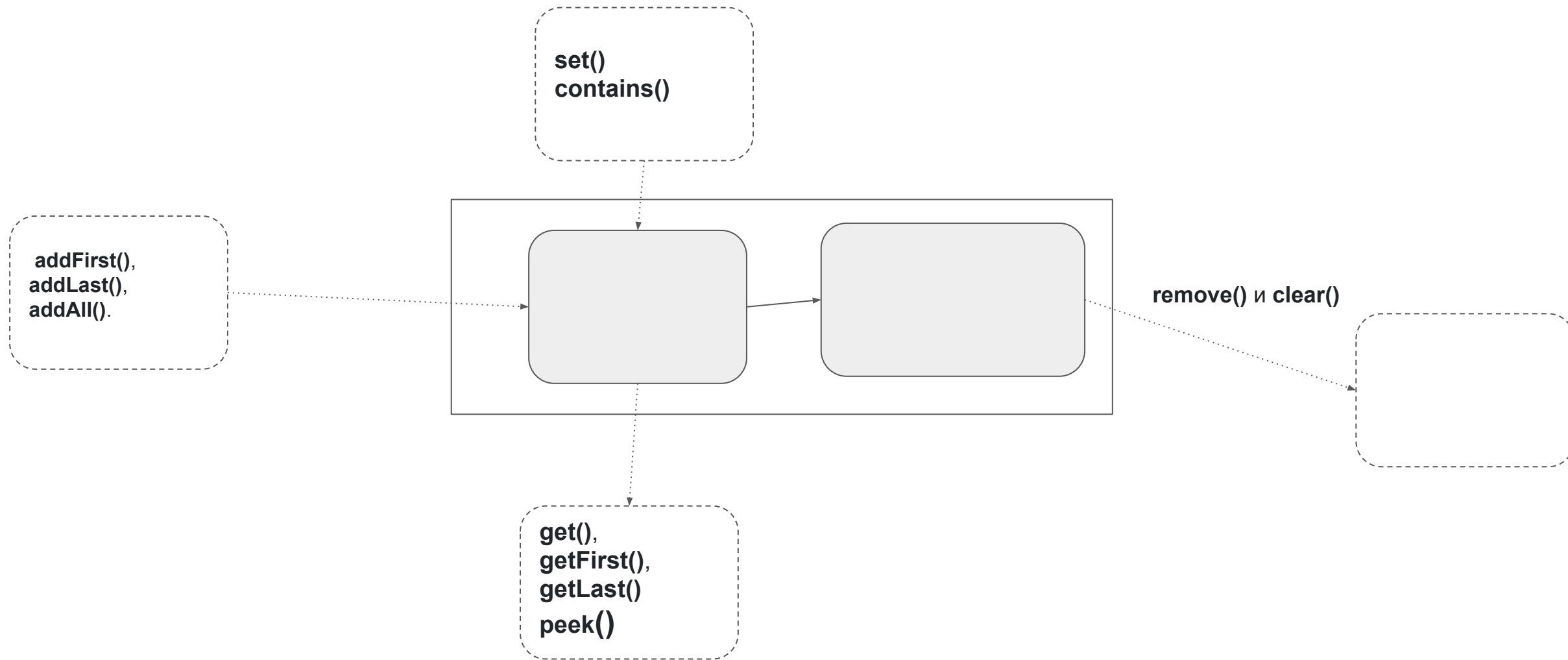
Удалим элемент со значением 5 из представленного



size	4
first	
last	

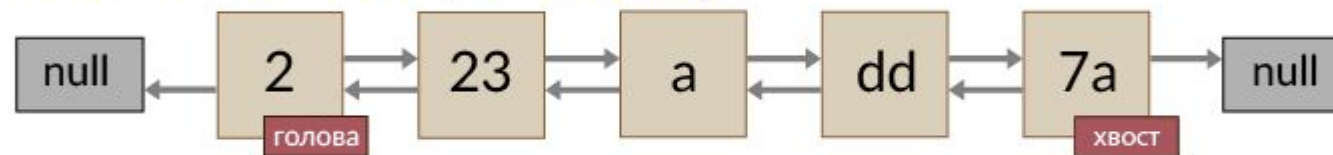
Рис. 8: Состояние связанного списка перед удалением элемента

Операции над *LinkedList*

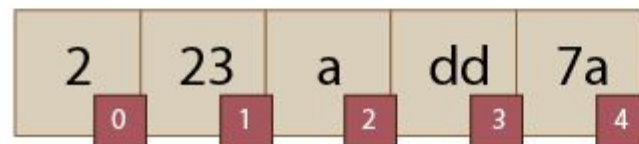


ArrayList vs. LinkedList

Связный список (LinkedList)



Массив (Array и ArrayList)



	Временная сложность							
	Среднее				Худшее			
	Индекс	Поиск	Вставка	Удаление	Индекс	Поиск	Вставка	Удаление
ArrayList	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Vector	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
LinkedList	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Hashtable	n/a	$O(1)$	$O(1)$	$O(1)$	n/a	$O(n)$	$O(n)$	$O(n)$
HashMap	n/a	$O(1)$	$O(1)$	$O(1)$	n/a	$O(n)$	$O(n)$	$O(n)$
LinkedHashMap	n/a	$O(1)$	$O(1)$	$O(1)$	n/a	$O(n)$	$O(n)$	$O(n)$
TreeMap	n/a	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	n/a	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$
HashSet	n/a	$O(1)$	$O(1)$	$O(1)$	n/a	$O(n)$	$O(n)$	$O(n)$
LinkedHashSet	n/a	$O(1)$	$O(1)$	$O(1)$	n/a	$O(n)$	$O(n)$	$O(n)$
TreeSet	n/a	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	n/a	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$