Skillbox

Курс «Java разработчик с нуля»

Основные коллекции и их устройство

Спикер: Шибков Константин

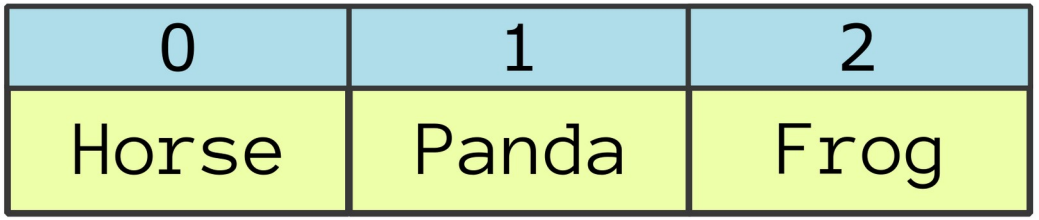
19.02.2021 16:00 МСК

Массивы

```
String[] animals = {“Horse”, “Panda”, “Frog”};
```

Skillbox

Массивы




```
String[] animals = new String[3];

animals[0] = "Horse";
animals[1] = "Panda";
animals[2] = "Frog";
```

Skillbox

Массивы

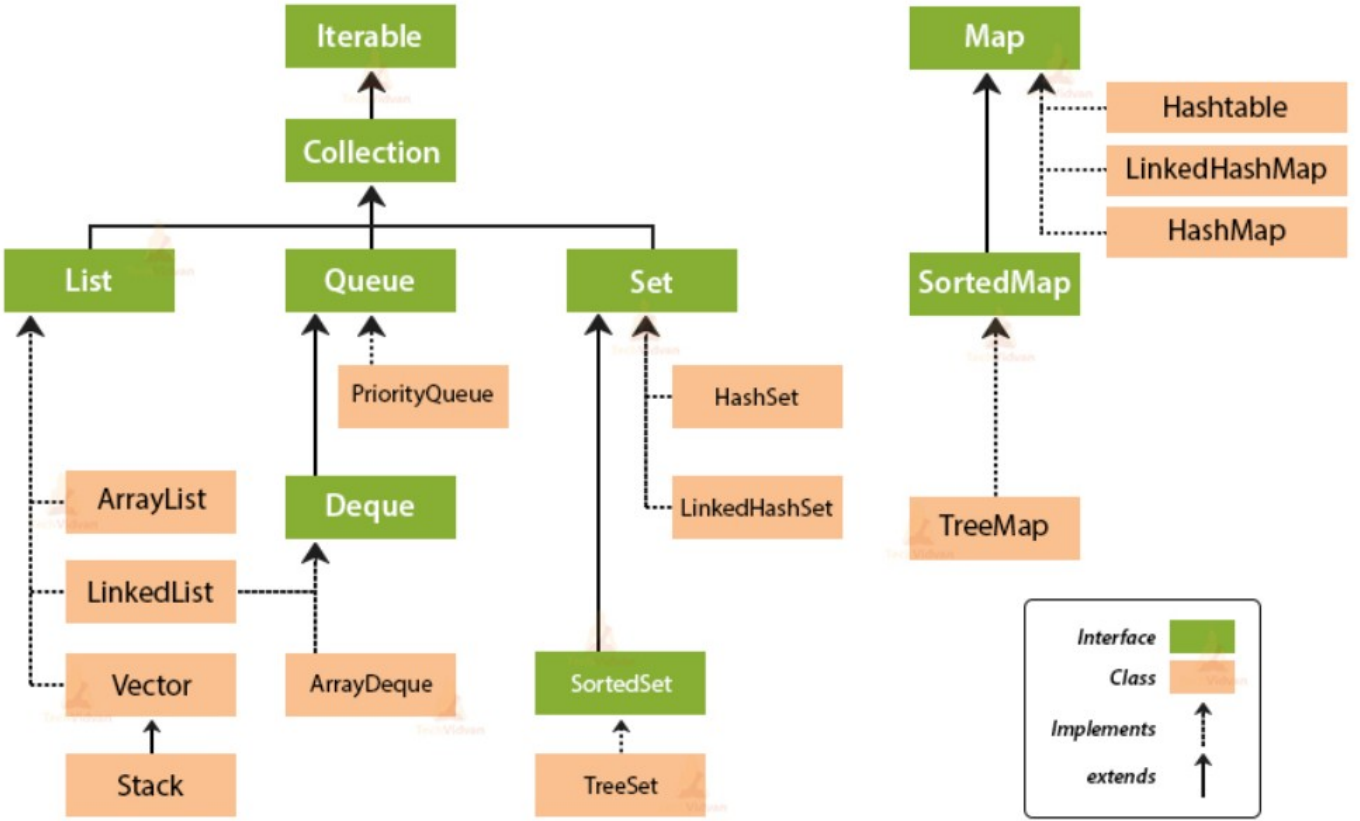

```
String[] animals = new String[3];

animals[0] = "Horse";
animals[1] = "Panda";
animals[2] = "Frog";

System.out.println(animals.length); //3
System.out.println(animals[2]); //Frog
```

Skillbox

Массивы

- количество ячеек (длину массива) невозможно поменять после создания
- обращения к содержимому ячейки происходит по индексу
- первый элемент массива имеет индекс со значением 0
- при объявлении, массив заполняется значениями по умолчанию (для объектов это null)
- если нам надо уменьшить и добавить элементы, это можно сделать только создав новый массив и переписав все значения в него из старого

Иерархия коллекций Java



Иерархия коллекций Java

Интерфейс Collection, обязывает все коллекции наследники выполнять следующие действия:

`size()` - получает размер коллекции
`add()` / `addAll()` - добавлять элемент или множества элементов в коллекцию
`contains()` - проверять если ли элемент в коллекции, если весть вернет true
`isEmpty()` - проверять есть в коллекции элементы, если нет то вернет true
`remove()` - удаляет переданный объект из коллекции

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collection.html

Иерархия коллекций Java

Интерфейс List, расширяет интерфейс Collection и позволяет и описывает работы со списками. В списке хранятся только объекты, сортировка отсутсвует, могут хранится одинаковые объекты.

Объявляет новые методы:
get(int idex) - обращается к элементам по индексу (как в массивах)
remove(int index) – удаляет по индексу элемента
indexOf(Object object) / lastIndexOf(Object object) - ищет индексы элемента
set(int index, E element) – заменяет значение элемента

Основные реализации: ArrayList и LinkedList

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/List.html

java.util.**ArrayList**

Skillbox

Коллекции - ArrayList

ArrayList - список, скрывающий внутри себя массив.

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/ArrayList.html

Skillbox

Коллекции - ArrayList

- данные хранятся в обычном массиве
- при создании списка внутри создается пустой массив на 10 элементов
- при добавлении элемента заполняются пустые ячейки массива
- если при попытке добавить элемент, и при этом массив заполнен:
  - создается новый массив на (n + n/2) элементов
  - данные из старого массива копируются в новый
  - добавляется новый элемент в расширенный массив

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/ArrayList.html

Skillbox

Коллекции – ArrayList - Параметры

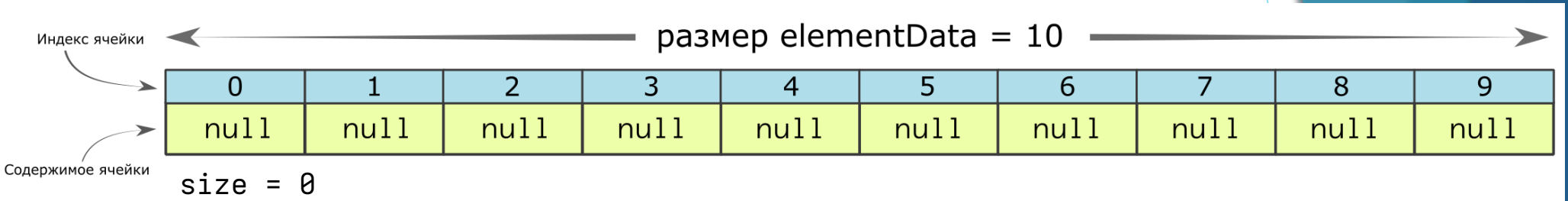
```
public class ArrayList<E> extends AbstractList<E>
        implements List<E>, RandomAccess, Cloneable, java.io.Serializable
{

    transient Object[] elementData;

    private int size;
```

* transient (англ. временный) – указанная переменная не будет сериализована.
https://stackoverflow.com/questions/9848129/why-does-arraylist-use-transient-storage

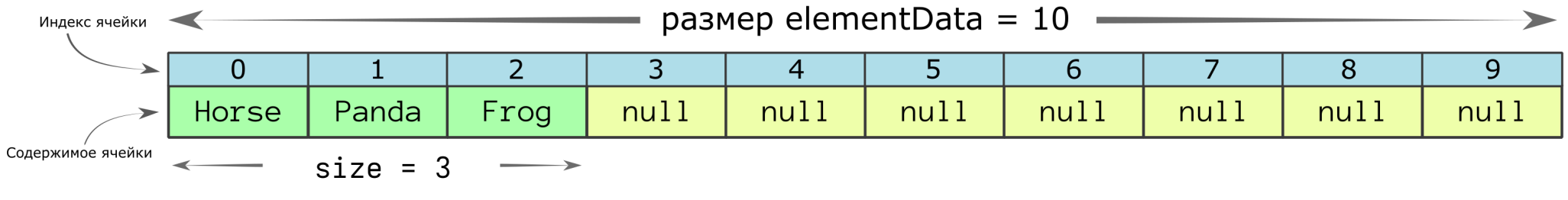
Skillbox

Коллекции – ArrayList – Добавление элементов

При создании списка используя конструктор по умолчанию:



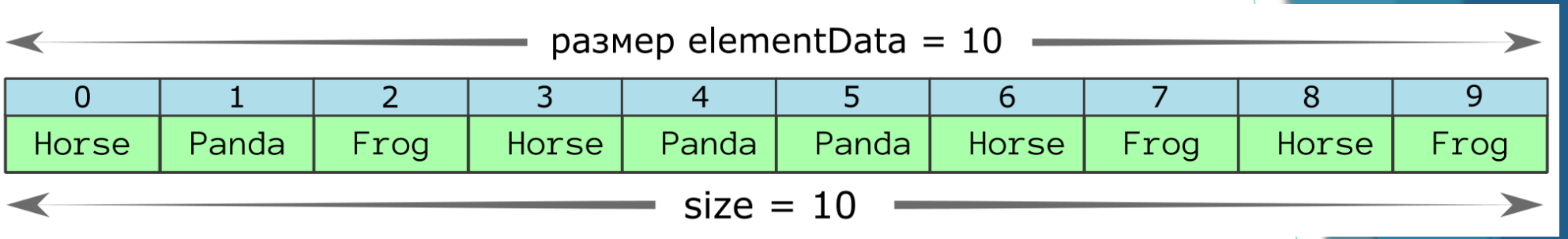
После добавления трех элементов в список:



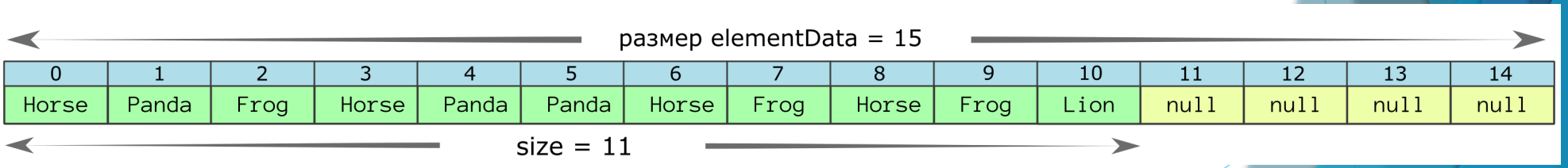
Skillbox

Коллекции – ArrayList – Добавление элементов

Состояния списка когда у нас заполнены все ячейки массива:



После добавления одиннадцатого элемента:

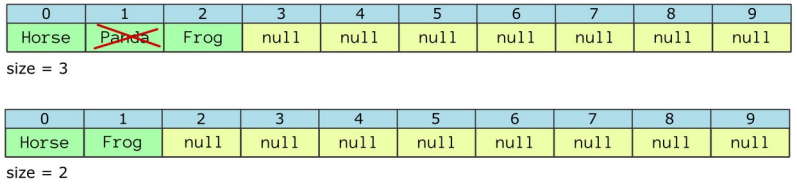


Skillbox

Коллекции – ArrayList – Удаление элементов

При удалении:

- одного элемента – все элементы справа сдвигаются левее



- список не уменьшает автоматически размер массива:

- при добавлении 2000 элементов в ArrayList, размер elementData[] будет 2776

- при удалении 1900 элементов, размер elementData[] останется 2776

- для принудительного сжатия массива, у класса ArrayList необходимо вызывать метод trimSize()

Коллекции – ArrayList – Сложность операций

- доступ к произвольному элементу по индексу за *константное* время *O(1)*;
- доступ к элементам по значению за *линейное* время *O(N)*;
- вставка в конец в среднем производится за *константное* время *O(1)*;
- удаление произвольного элемента из списка занимает значительное время т.к. при этом все элементы, находящиеся «правее» смещаются на одну ячейку влево (реальный размер массива (capacity) не изменяется);
- вставка элемента в произвольное место списка занимает значительное время т.к. при этом все элементы, находящиеся «правее» смещаются на одну ячейку вправо;

Skillbox

public ArrayList(Collection<? extends E> c) {

Коллекции – ArrayList - Конструкторы

- При создании ArrayList возможно задать первоначальный размер массива elementData вызвав конструктор:

```
ArrayList<String> list = new ArrayList<>(80);
```

где 80 это Capacity, первоначальный размер elementData. Далее увеличение происходит по такому же правилу (n+n/2): 120, 180, 270...

- Последний конструктор позволяет создать ArrayList на основе другой коллекции:

```
public ArrayList(Collection<? extends E> c) {
```

Передать можно любую коллекцию наследующую Collection

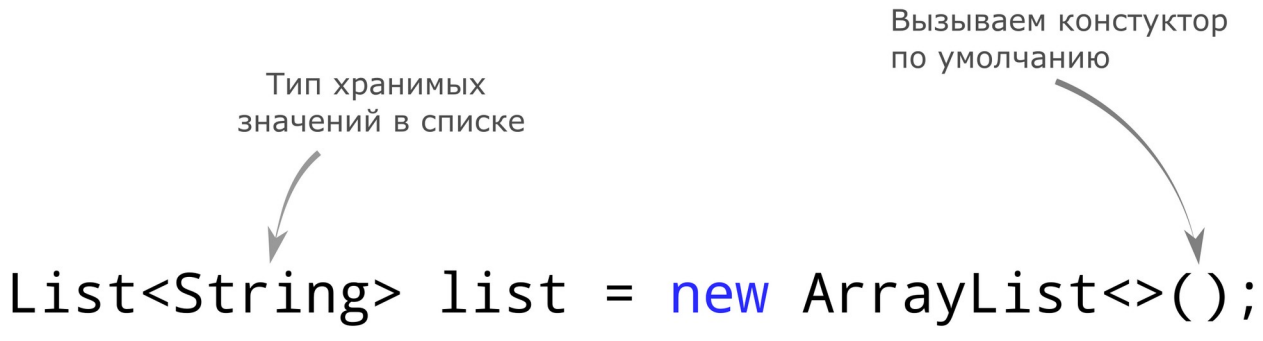
Skillbox

java.util.**LinkedList**

Skillbox

Коллекции - LinkedList

LinkedList - список, где один элемент ссылается на соседний.



https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/ArrayList.html

Skillbox

Коллекции – LinkedList - Параметры

```
public class LinkedList<E>
    extends AbstractSequentialList<E>
    implements List<E>, Deque<E>, Cloneable, java.io.Serializable
{
    transient int size = 0;

    // Pointer to first node.
    transient Node<E> first;

    //Pointer to last node.
    transient Node<E> last;
```

* transient (англ. временный) – указанная переменная не будет сериализована.
https://stackoverflow.com/questions/9848129/why-does-arraylist-use-transient-storage

Skillbox

Коллекции – LinkedList – Структура

Данные не хранятся в самом списке, с списке только ссылки на первый и последний элемент списка (Node). Данные каждого элемента хранятся в самих Node.

```
public class LinkedList<E>
    extends AbstractSequentialList<E>
    implements List<E>, Deque<E>, Cloneable, java.io.Serializable
{
    transient int size = 0;

    // Pointer to first node.
    transient Node<E> first;

    //Pointer to last node.
    transient Node<E> last;
```

* transient (англ. временный) – указанная переменная не будет сериализована.
https://stackoverflow.com/questions/9848129/why-does-arraylist-use-transient-storage

Skillbox

Коллекции – LinkedList – Структура

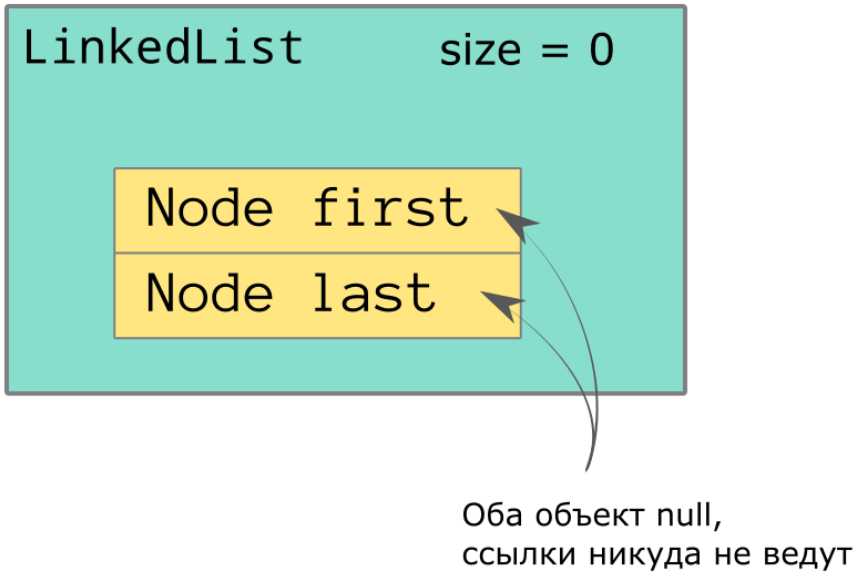
Каждый элемент списка (Node) имеет ссылку на предыдущий и следующий элемент.

```
class Node<E> {

    E item;

    Node<E> next;

    Node<E> prev;
```

Skillbox

Коллекции – LinkedList – Структура

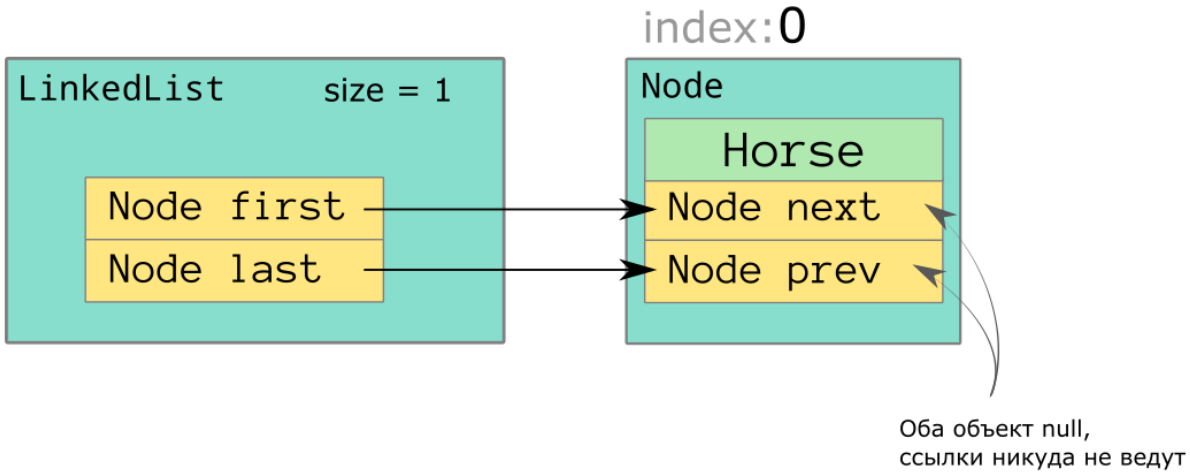
Вновь созданный LinkedList содержит ссылки который не ведут на другие Node.



Skillbox

Коллекции – LinkedList – Структура

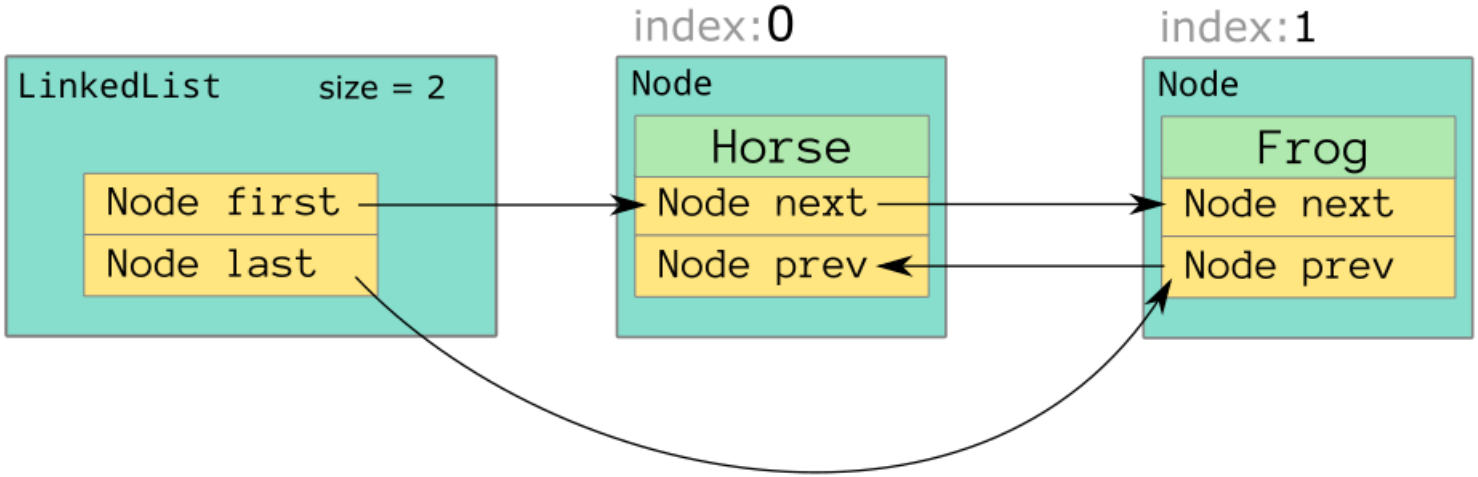
Когда мы добавляем один элемент через метод add(), происходит заполнение ссылок, обе ссылки в этот раз будут указывать на добавленный объект.



Skillbox

Коллекции – LinkedList – Структура

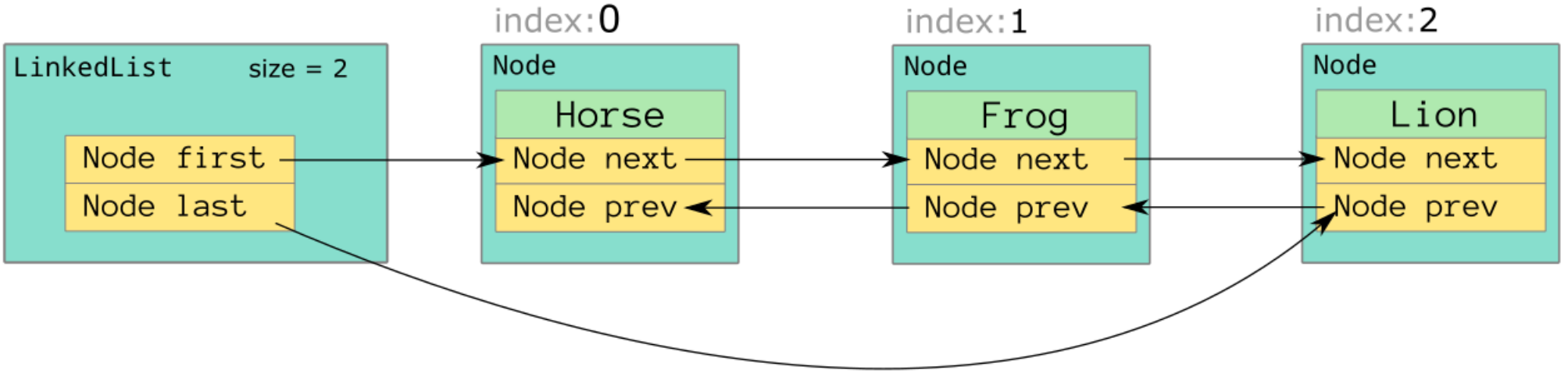
Добавляя еще один элемент мы меняем значение last в списке, и назначаем next у элемента с индексом 0 “Horse” на новый элемент “Frog”



Skillbox

Коллекции – LinkedList – Структура

Аналогично производятся и остальные добавления. И таким образом у нас получается такой список, который мы может обойти как в прямом от first -> next, так и в обратном направлении last -> prev.

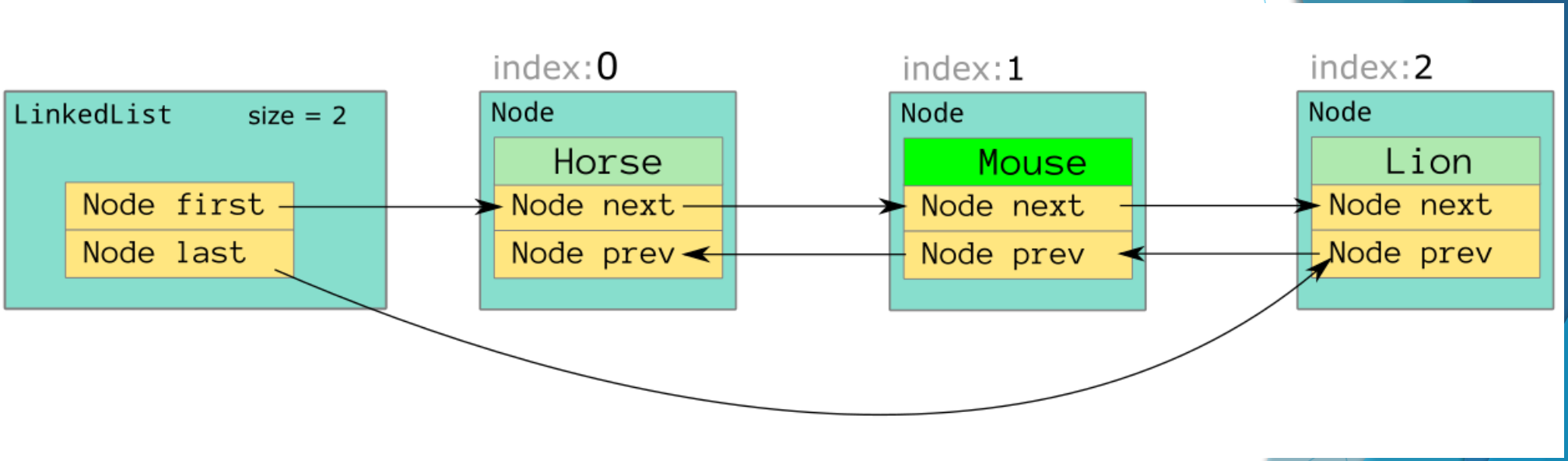


Это позволяем нам быстрее добраться до нужного элемента, мы не можем напрямую обратится под индексу, нам надо проходить каждый элемент до того который нам нужен. Зная индекс, мы рассчитываем короткий путь, нам надо с конца начинать или с начала, поэтому количество итераций для получения элемента по индексу это N/2, где N количество элементов.

Skillbox

Коллекции – LinkedList – Структура

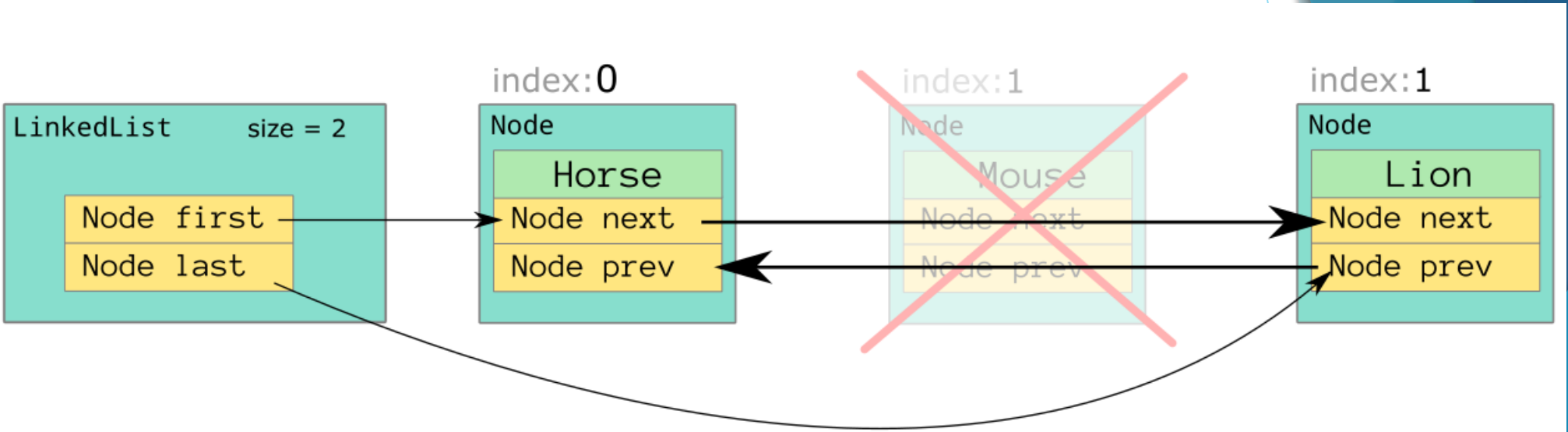
Для изменения данных set(1, “Mouse”) нам надо найти нужный элемент по индексу и заменить содержимое, при этом связи остаются.



Skillbox

Коллекции – LinkedList – Структура

Для удаления элемента, соседи меняют ссылки.



Skillbox

Подробнее про переполнение и варианты предотвращения:

https://wiki.sei.cmu.edu/confluence/display/java/NUM00-J.+Detect+or+prevent+integer+overflow

Расчеты с использованием double, float. Причины неточности таких расчетов:

https://habr.com/ru/post/219595/

https://habr.com/ru/post/112953/

Перегрузка методов:

https://habr.com/ru/company/otus/blog/428307/

Heap и stack:

https://javadevblog.com/chto-takoe-heap-i-stack-pamyat-v-java.html