Основы программирование: Введение в Java

Лекция 10. Коллекции.

Власенко Олег Федосович

Пример

Необходимо на основе текстового файла «Текст» создать файл со словарем «Словарь».

Пример:

«Текст»:

Мама мыла раму. Мама мыла яблоко

«Словарь»:

мама

мыла

раму

яблоко

Пример (идея реализации)

Алгоритм:

1). Читаем файл «Текст» посимвольно, вычленяя слова из текста.

2). Каждое слово добавляем в КОЛЛЕКЦИЮ, если его там не было до этого.

3). Коллекцию сохраняем в файле «Словарь»

Пример: коллекция Vector

Collection dict = new **Vector();**

1) слово читается из файла «Текст»

String word = ...;

2) Слово добавляется в коллекцию, если его там не было

```
if (!dict.contains(word)) {
        dict.add(word);
}
```

3) Коллекция сохраняется в файле

```
BufferedWriter bw =  new BufferedWriter(...);
for (Object word : dict) {
        bw.write((String)word);
        bw.newLine();
}
bw.close();
```

Пример: коллекция ArrayList

```
Collection dict = new ArrayList();
1) слово читается из файла «Текст»
String word = ...;
2)  Слово добавляется в коллекцию, если его там не было
if (!dict.contains(word)) {
        dict.add(word);
}
3) Коллекция сохраняется в файле
BufferedWriter bw =  new BufferedWriter(...);
for (Object word : dict) {
        bw.write((String)word);
        bw.newLine();
}
bw.close();
```

Чем отличаются Vector и ArrayList?

http://www.quizful.net/interview/java/vector-arraylist-difference

Вопрос

В чем принципиальное отличие классов Vector и ArrayList

Ответ

Методы класса Vector синхронизированы, в то время как ArrayList - нет.

Потоки и синхронизация

- Коротко про потоки
- Коротко про синхронизацию

Разбор кода примера

- DictLoaderTest
- DictLoader.saveDictToFile
  - Работа с файлами при записи

- DictLoader.loadTextToDict
  - Разбор алгоритма
    - Флаги – inWord

Обработка исключений

- Exception
  - *IOException*
    - FileNotFoundException
  - Правила перехвата с учетом иерархии классов исключений

Демонстрация работы

```
Collection dict = new LinkedList();
```

vs

```
Collection dict = new HashSet();
```

Пример: коллекция TreeSet

Collection dict = new **TreeSet();**

1) слово читается из файла «Текст»

String word = ...;

2) Слово добавляется в коллекцию, если его там не было

```
if (!dict.contains(word)) {
        dict.add(word);
}
```

3) Коллекция сохраняется в файле

```
BufferedWriter bw =  new BufferedWriter(...);
for (Object word : dict) {
        bw.write((String)word);
        bw.newLine();
}
bw.close();
```

Чем отличаются Vector и TreeSet?

Время работы с Vector: 758 мс

Время работы с TreeSet: 104 мс

Пример:
разные коллекции

Collection dict = new **Vector();**

Collection dict = new **ArrayList();**

Collection dict = new **LinkedList();**

Collection dict = new **TreeSet();** // элементы упорядочены
// по значению, но не по порядку добавления

Collection dict = new **HashSet();** // порядок обхода неопределен

Collection dict = new **LinkedHashSet();** // порядок обхода
// элементов = порядку добавления элементов

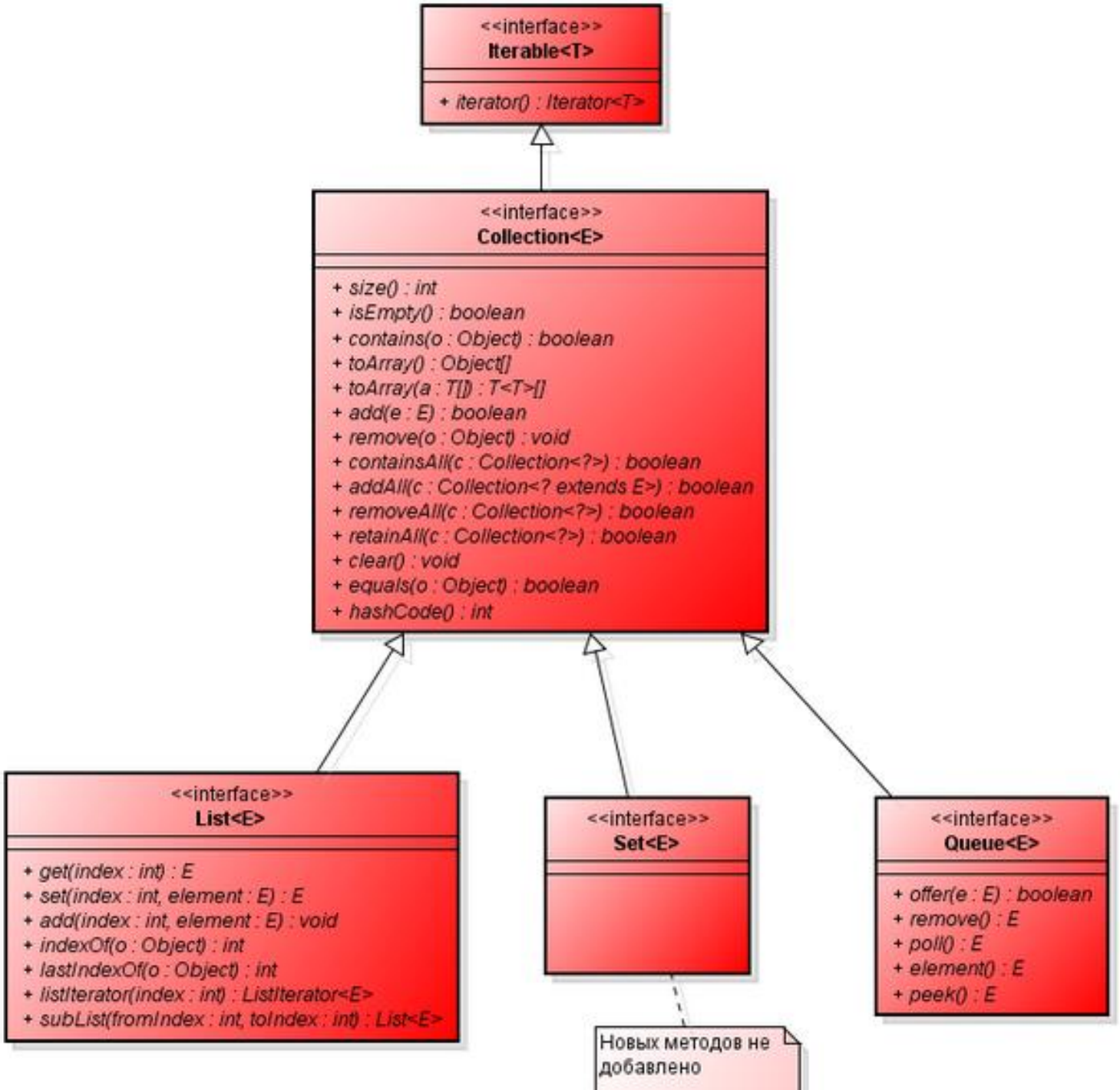
```
1) слово читается из файла «Текст»
String word = ...;
2)  Слово добавляется в коллекцию, если его там не было
if (!dict.contains(word)) {
        dict.add(word);
}
3) Коллекция сохраняется в файле
BufferedWriter bw =  new BufferedWriter(...);
for (Object word : dict) {
        bw.write((String)word);
        bw.newLine();
}
bw.close();
```

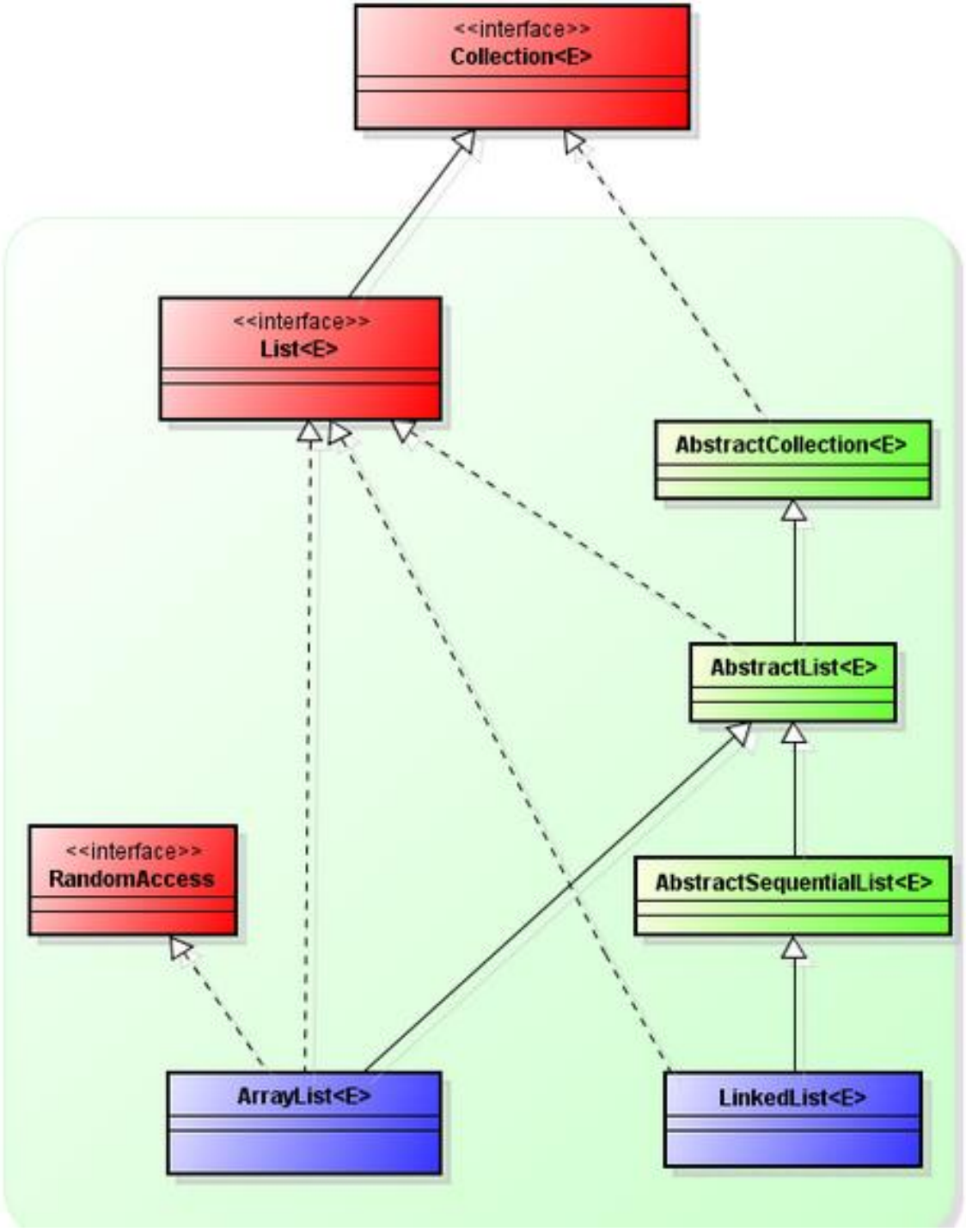
Пример:
разные коллекции – время работы

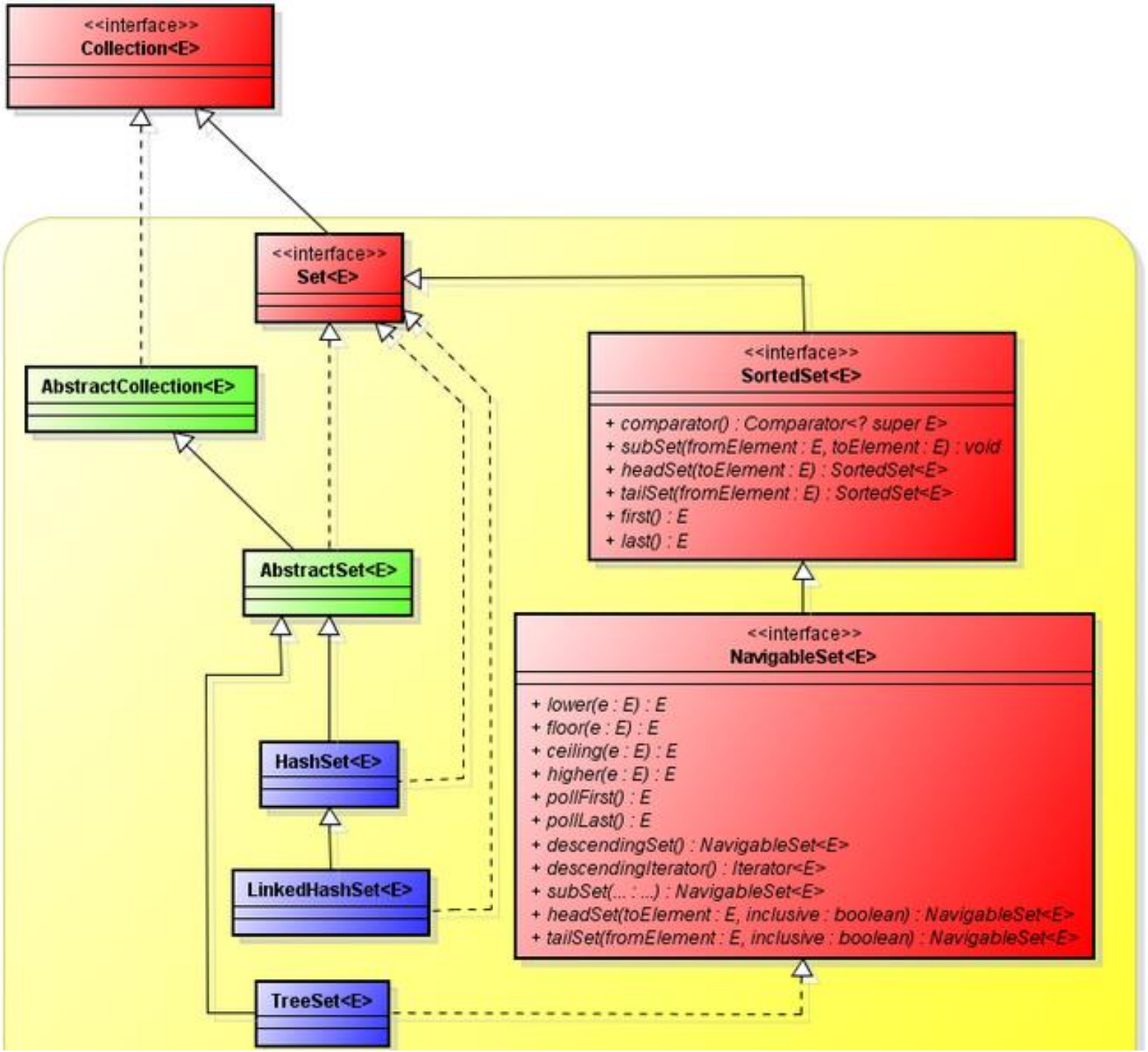
	170 Kb	4,4 Mb
ArrayList	0.713	16.151
LinkedList	0.753	18.828
Vector	0.697	16.020
TreeSet	0.059	0.532
LinkedHashSet	0.051	0.397
HashSet	0.044	0.372

Иерархии коллекций

- Источник: http://www.quizful.net/post/Java-Collections


<<interface>>
Iterable<T>
+ iterator() : Iterator<T>
<<interface>>
Collection<E>
+ size() : int
+ isEmpty() : boolean
+ contains(o : Object) : boolean
+ toArray() : Object[]
+ toArray(a : T[]) : T<T>[]
+ add(e : E) : boolean
+ remove(o : Object) : void
+ containsAll(c : Collection<?>) : boolean
+ addAll(c : Collection<? extends E>) : boolean
+ removeAll(c : Collection<?>) : boolean
+ retainAll(c : Collection<?>) : boolean
+ clear() : void
+ equals(o : Object) : boolean
+ hashCode() : int
<<interface>>
List<E>
+ get(index : int) : E
+ set(index : int, element : E) : E
+ add(index : int, element : E) : void
+ indexOf(o : Object) : int
+ lastIndexOf(o : Object) : int
+ listIterator(index : int) : ListIterator<E>
+ subList(fromIndex : int, toIndex : int) : List<E>
<<interface>>
Set<E>
Новых методов не добавлено
<<interface>>
Queue<E>
+ offer(e : E) : boolean
+ remove() : E
+ poll() : E
+ element() : E
+ peek() : E


<<interface>>
Collection<E>
<<interface>>
List<E>
AbstractCollection<E>
AbstractList<E>
<<interface>>
RandomAccess
AbstractSequentialList<E>
ArrayList<E>
LinkedList<E>


<<interface>>
Collection<E>
<<interface>>
Set<E>
AbstractCollection<E>
<<interface>>
SortedSet<E>
+ comparator() : Comparator<? super E>
+ subSet(fromElement : E, toElement : E) : void
+ headSet(toElement : E) : SortedSet<E>
+ tailSet(fromElement : E) : SortedSet<E>
+ first() : E
+ last() : E
AbstractSet<E>
<<interface>>
NavigableSet<E>
+ lower(e : E) : E
+ floor(e : E) : E
+ ceiling(e : E) : E
+ higher(e : E) : E
+ pollFirst() : E
+ pollLast() : E
+ descendingSet() : NavigableSet<E>
+ descendingIterator() : Iterator<E>
+ subSet(... : ...) : NavigableSet<E>
+ headSet(toElement : E, inclusive : boolean) : NavigableSet<E>
+ tailSet(fromElement : E, inclusive : boolean) : NavigableSet<E>
HashSet<E>
LinkedHashSet<E>
TreeSet<E>

Итераторы

public interface Iterator<E>

Modifier and Type	Method and Description
boolean	hasNext() Returns true if the iteration has more elements.
E	next() Returns the next element in the iteration.
void	remove() Removes from the underlying collection the last element returned by this iterator (optional operation).

Задача 1 – Знакомство с ArrayList

Создать коллекцию ArrayList. Добавить в нее 3 элемента типа String. Вывести элементы, содержащиеся в коллекции в консоль при помощи итератора.

Задача 1 – Решение

```
ArrayList v = new ArrayList();
v.add("10");
v.add("2");
v.add("30");

Iterator iterator = v.iterator();
while (iterator.hasNext()) {
        Object obj = iterator.next();
        System.out.println(obj);
}
```

Класс Object

Modifier and Type	Method and Description
protected Object	clone() Creates and returns a copy of this object.
boolean	**equals(Object obj)** Indicates whether some other object is "equal to" this one.
protected void	finalize() Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
Class<?>	getClass() Returns the runtime class of this Object.
int	**hashCode()** Returns a hash code value for the object.

Класс Object (2)

Modifier and Type	Method and Description
void	notify() Wakes up a single thread that is waiting on this object's monitor.
void	notifyAll() Wakes up all threads that are waiting on this object's monitor.
String	**toString()** Returns a string representation of the object.

Класс Object

Modifier and Type	Method and Description
void	wait() Causes the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.
void	wait(long timeout) Causes the current thread to wait until either another thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has elapsed.
void	wait(long timeout, int nanos) Causes the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object, or some other thread interrupts the current thread, or a certain amount of real time has elapsed.

Использование toString()

```
ArrayList a3 = new ArrayList();
a3.add("Один");
a3.add("Два");
a3.add("Пять");

System.out.println("a3 = " + a3);
```

Коротко о структурах данных

- Динамический массив
- Список
- Хэш
- Двоичное дерево поиска

Хэш

Пример про заказы по телефону

Какие решения возможны?

Как искать заказ за 5 секунд вручную?

Структура хэша:

1. Хэш-функция
2. Хэш-таблица
3. Схема разрешения коллизий (список)

Спасибо за внимание!

Власенко Олег Федосович

E-mail: vlasenko.oleg@gmail.com

Vk: vk.com/oleg.f.vlasenko

Телефон: 8 902 246 05 47