

Java Professional module #2

lecture #5. Java Set implementation. Exercises.

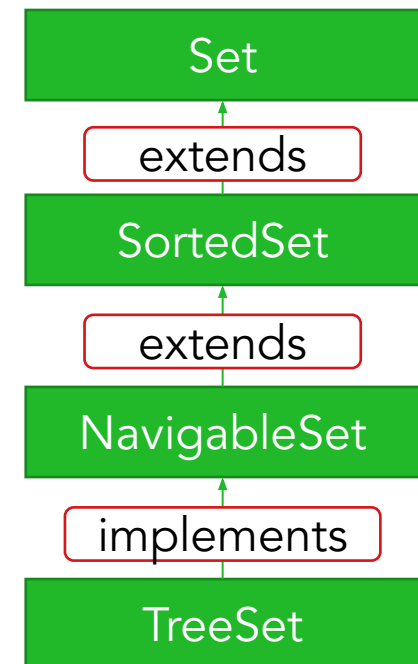
Mentor:

lecture #5. Java Set implementation. Exercises.

- Set in Java
- Set Objects
- Operations on the Set Interface
- HashSet basic operations
- Classes that implement the Set interface
- Array to Set

Java Set

- Интерфейс set присутствует в пакете java.util и расширяет интерфейс Collection.
- Set **неупорядоченная коллекция объектов**, в которой **не могут** храниться повторяющиеся значения.
- Set интерфейс содержит методы, унаследованные от интерфейса Collection, и добавляет функцию, которая **ограничивает** вставку **повторяющихся** элементов.
- Есть два интерфейса, которые расширяют реализацию Set:
 - SortedSet
 - NavigableSet



Interface SortedSet

- расширяет интерфейс Set
- интерфейс содержит методы, унаследованные от интерфейса Set, и добавляет функцию, которая хранит все элементы этого интерфейса в отсортированном виде
- Все элементы SortedSet должны реализовывать интерфейс Comparable или быть приняты Comparator по умолчанию
- SortedSet — это интерфейс , нельзя создавать объекты типа SortedSet, всегда нужен класс!
- TreeSet — это класс, реализующий интерфейс SortedSet

Interface NavigableSet

- Интерфейс NavigableSet наследуется от интерфейса SortedSet.
- NavigableSet ведет себя как SortedSet, за исключением того, что у нас есть методы навигации, доступные в дополнение к механизмам сортировки SortedSet.
- Например, интерфейс NavigableSet может перемещаться по набору в обратном порядке по сравнению с порядком, определенным в SortedSet.
- Доступ к NavigableSet и его перемещение можно осуществлять как в порядке возрастания, так и в порядке убывания.
- Классы, реализующие этот интерфейс: TreeSet и ConcurrentSkipListSet.

TreeSet и ConcurrentSkipListSet

- TreeSet — это, по сути, реализация самобалансирующегося бинарного дерева поиска
 - TreeSet служит отличным выбором для хранения больших объемов отсортированной информации, к которой предполагается быстрый доступ.
 - *добавление, удаление и поиск, занимают время $O(\log(N))$*
 - Вставка null значений в TreeSet вызывает исключение NullPointerException, потому что при вставке нулевого значения оно сравнивается с существующими элементами, а null нельзя сравнивать ни с каким значением.
- ConcurrentSkipListSet параллельная версия NavigableSet.
- Элементы в ConcurrentSkipListSet сортируются по умолчанию в их естественном порядке или с помощью компаратора.
- Поскольку ConcurrentSkipListSet реализует SortedSet<E> и NavigableSet<E> , то похож на TreeSet с добавленной функцией одновременности.
- ConcurrentSkipListSet является **потокобезопасным** и может использоваться несколькими потоками одновременно
- TreeSet **НЕ** является потокобезопасным.

HashSet in Java

- Класс HashSet реализует интерфейс Set
- Не гарантируется порядок итерации набора
- Этот класс разрешает null элемент
- HashSet – $O(1)$ для основных операций, таких как добавление, удаление, содержание и размер, при условии, что хеш-функция правильно распределяет элементы
- Базовой структурой данных для HashSet является Hashtable
- повторяющиеся значения не допускаются
- Объекты вставляются на основе их хэш-кода
- подкласс HashSet — LinkedHashSet

HashSet in Java

Методы HashSet:

1. `boolean add(E e)`: добавляет элемент в HashSet, если таковой отсутствует, если же такой элемент уже присутствует, метод возвращает `false`.
2. `void clear()`: удаляет все элементы.
3. `boolean contains(Object o)`: Возвращает `true`, если данный элемент присутствует.
4. `boolean remove(Object o)`: удаляет данный элемент, если таковой присутствует.
5. `Iterator iterator()`: возвращает итератор для элементов.
6. `boolean isEmpty()`: возвращает `true`, если нет элементов.
7. `Object clone()`: выполняет поверхностное клонирование HashSet.

Array to Set

Способ 1: Наивный метод

Способ 2: Использование метода `Arrays.asList()`

Способ 3: Использование `Collections.addAll()`

Способ 4: использование Java 8 Stream API

Способ 5: Использование метода `Set.of()`