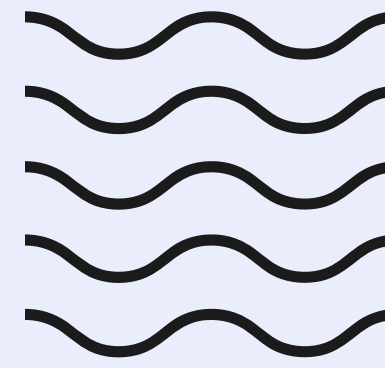


TreeMap



ПОДГОТОВИЛ: АЛИШЕР ХАМИДОВ



Основные моменты:

План лекции

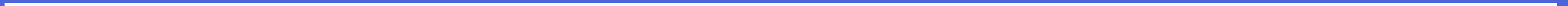
1. SortedMap
2. NavigableMap
3. Внутреннее устройство





Что такое TreeMap?

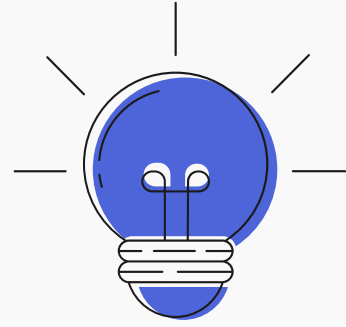
Ваши ассоциации





Чаще всего мы используем такую имплементацию Map как HashMap.

Иногда необходимо хранить данные в структурированном виде с возможностью навигации по ним. В таком случае на помощь приходит другая реализация интерфейса Map — TreeMap



TreeMap реализует интерфейсы SortedMap и NavigableMap

SortedMap — интерфейс, который расширяет Map и добавляет методы, актуальные для отсортированного набора данных:

- **firstKey()**: возвращает ключ первого элемента карты;
- **lastKey()**: возвращает ключ последнего элемента;
- **headMap(K end)**: возвращает map, которая содержит все элементы текущей, от начала до элемента с ключом end;
- **tailMap(K start)**: возвращает map, которая содержит все элементы текущей, начиная с элемента start и до конца;
- **subMap(K start, K end)**: возвращает map, которая содержит все элементы текущей, начиная с элемента start и до элемента с ключом end.

NavigableMap — интерфейс, который расширяет SortedMap и добавляет методы для навигации между элементами:

- **firstEntry()**: возвращает первую пару “ключ-значение”;
- **lastEntry()**: возвращает последнюю пару “ключ-значение”;
- **pollFirstEntry()**: возвращает и удаляет первую пару;
- **pollLastEntry()**: возвращает и удаляет последнюю пару;
- **ceilingKey(K obj)**: возвращает наименьший ключ k, который больше или равен ключу obj. Если такого ключа нет, возвращает null;
- **floorKey(K obj)**: возвращает самый большой ключ k, который меньше или равен ключу obj. Если такого ключа нет, возвращает null;
- **lowerKey(K obj)**: возвращает наибольший ключ k, который меньше ключа obj. Если такого ключа нет, возвращает null;

NavigableMap — интерфейс, который расширяет SortedMap и добавляет методы для навигации между элементами:

- **higherKey(K obj)**: возвращает наименьший ключ k, который больше ключа obj. Если такого ключа нет, возвращает null;
- **ceilingEntry(K obj)**: аналогичен методу ceilingKey(K obj), только возвращает пару “ключ-значение” (или null);
- **floorEntry(K obj)**: аналогичен методу floorKey(K obj), только возвращает пару “ключ-значение” (или null);
- **lowerEntry(K obj)**: аналогичен методу lowerKey(K obj), только возвращает пару “ключ-значение” (или null);
- **higherEntry(K obj)**: аналогичен методу higherKey(K obj), только возвращает пару “ключ-значение” (или null);

NavigableMap — интерфейс, который расширяет SortedMap и добавляет методы для навигации между элементами:

- **descendingKeySet()**: возвращает NavigableSet, содержащий все ключи, отсортированные в обратном порядке;
- **descendingMap()**: возвращает NavigableMap, содержащую все пары, отсортированные в обратном порядке;
- **navigableKeySet()**: возвращает объект NavigableSet, содержащий все ключи в порядке хранения;
- **headMap(K upperBound, boolean incl)**: возвращает мапу, которая содержит пары от начала и до элемента upperBound. Аргумент incl указывает, нужно ли включать элемент upperBound в возвращаемую мапу;

NavigableMap — интерфейс, который расширяет SortedMap и добавляет методы для навигации между элементами:

- **tailMap(K lowerBound, boolean incl)**: функционал похож на предыдущий метод, только возвращаются пары от lowerBound и до конца;
- **subMap(K lowerBound, boolean lowIncl, K upperBound, boolean highIncl)**: как и в предыдущих методах, возвращаются пары от lowerBound и до upperBound, аргументы lowIncl и highIncl указывают, включать ли граничные элементы в новую mapу.



Важное замечание о TreeMap

В этой реализации объекты сортируются и сохраняются в порядке возрастания в соответствии с их естественным порядком. TreeMap использует самобалансирующееся бинарное дерево поиска, точнее красно-черное дерево.



Важное замечание о TreeMap

Проще говоря, будучи самобалансирующимся бинарным деревом поиска, каждый узел бинарного дерева содержит дополнительный бит, который используется для определения цвета узла: красного или черного. Во время последующих вставок и удалений эти «цветные» биты помогают обеспечить более или менее сбалансированное дерево.

Узнать больше о красно-черном дереве вы можете по ссылкам:

<https://habr.com/ru/company/otus/blog/472040/>

<https://www.baeldung.com/cs/red-black-trees>

