

## Bài 5.8: Tìm hiểu lớp TreeMap

---

- ✓ Khái niệm và đặc điểm
- ✓ Các phương thức và mô tả
- ✓ Ví dụ minh họa

# Khái niệm và đặc điểm

- Là một triển khai của NavigableMap dựa trên cây đỏ đen.
- Cây đỏ đen là một loại cây nhị phân tìm kiếm tự cân bằng trong đó mỗi node có thêm một bit và bit đó thường được hiểu là màu(đỏ hoặc đen).
- Các key của map được sắp xếp theo thứ tự tự nhiên hoặc theo quy luật được chỉ định bởi comparator cho trước khi tạo đối tượng của TreeMap.
- Triển khai này đảm bảo cho các thao tác kiểm tra key, lấy, gán hay xóa phần tử được thực hiện với chi phí thời gian bằng hằng số.
- Triển khai này không đồng bộ.
- TreeMap là một phần của Java Collections FrameWork.

# Các phương thức và mô tả

Phương thức	Mô tả
TreeMap()	Khởi tạo một map mới rỗng, sử dụng thứ tự tự nhiên của các key của nó.
TreeMap(Comparator<? Super K> comparator)	Tạo một map mới rỗng, thứ tự sắp xếp dựa vào comparator cho trước.
TreeMap(Map<? Extends K, ? extends V> m)	Tạo một map mới từ các phần tử của một map m cho trước. Thứ tự phần tử được sắp đặt theo trật tự tự nhiên.
TreeMap(SortedMap<K, ? extends V> m)	Tạo một map mới từ một map cho trước, sử dụng cùng thứ tự sắp xếp thiết lập như trong map truyền vào.
Map.Entry<K, V> ceilingEntry(K key)	Trả về một ánh xạ key-value liên kết với key lớn hơn hoặc tương đương với key hoặc trả về null nếu không có key nào thỏa.
K ceilingKey(K key)	Trả về key nhỏ nhất lớn hơn key đã cho hoặc trả về null nếu không có key nào thỏa mãn.
void clear()	Trả về tất cả các ánh xạ từ map hiện tại.
Object clone()	Trả về một bản sao của đối tượng TreeMap.
Comparator<? Super K> comparator()	Trả về comparator sử dụng để sắp xếp key của map hiện thời. Trả về null nếu map hiện tại sử dụng thứ tự sắp xếp tự nhiên để sắp xếp key của nó.

# Các phương thức và mô tả

boolean containKey(Object key)	Trả về true nếu map chứa ánh xạ của key được chỉ định.
boolean containValue(Object value)	Trả về true nếu map chứa 1 hoặc nhiều ánh xạ đến giá trị value cho trước.
NavigableSet<K> descendingKeySet()	Trả về một NavigableSet với các key được đảo ngược thứ tự từ map hiện thời.
NavigableMap<K, V> descendingMap()	Trả về một NavigableMap đảo ngược thứ tự các ánh xạ có trong map hiện thời.
Set<Map.Entry<K, V>> entrySet()	Trả về một Set các ánh xạ chứa trong map hiện thời.
Map.Entry<K, V> firstEntry()	Trả về một ánh xạ key-value liên kết với key nhỏ nhất trong map hiện tại. Trả về null nếu map rỗng.
K firstKey()	Trả về key nhỏ nhất trong map hiện tại.
Map.Entry<K, V> floorEntry(K key)	Trả về một ánh xạ liên kết với key lớn nhất nhỏ hơn hoặc bằng key được chỉ định. Trả về null nếu không có key nào thỏa.
K floorKey(K key)	Trả về key lớn nhất nhỏ hơn hoặc bằng key được chỉ định. Trả về null nếu không có key thỏa.
V get(Object key)	Trả về giá trị tương ứng với key hoặc null nếu không tồn tại ánh xạ ứng với key được chỉ định.
SortedMap<K, V> headMap(K toKey)	Trả về một SortedMap trong đó các key nhỏ hơn toKey.

# Các phương thức và mô tả

Map.Entry<K, V> higherEntry(K key)	Trả về một ánh xạ liên kết với key nhỏ nhất lớn hơn key được chỉ định hoặc trả về null nếu không tồn tại key thỏa mãn.
K higherEntry(K key)	Trả về key nhỏ nhất lớn hơn key được chỉ định hoặc null nếu không tồn tại key thỏa mãn.
Set<K> keySet()	Trả về một Set các key có trong map hiện tại.
Map.Entry<K, V> lastEntry()	Trả về một ánh xạ liên kết với key lớn nhất trong map hoặc null nếu map rỗng.
K lastKey()	Trả về key lớn nhất trong map hiện thời.
Map.Entry<K, V> lowerEntry(K key)	Trả về một ánh xạ liên kết với key lớn nhất nhỏ hơn key được chỉ định hoặc trả về null nếu không tồn tại key thỏa mãn.
K lowerKey(K key)	Trả về key lớn nhất nhỏ hơn key được chỉ định hoặc null nếu không có key nào thỏa.
NavigableSet<K> NavigableKeySet()	Trả về một NavigableSet chứa các key của map hiện thời.
Map.Entry<K, V> pollFirstEntry()	Trả về ánh xạ liên kết với key nhỏ nhất trong map hiện thời. Trả về null nếu map rỗng.
Map.Entry<K, V> pollLastEntry()	Trả về ánh xạ liên kết với key lớn nhất trong map hiện thời. Trả về null nếu map rỗng.

# Các phương thức và mô tả

V put(K key, V value)	Liên kết giá trị value cho trước với key được chỉ định trong map.
void putAll(Map<? Extends K, ? extends V> map)	Sao chép tất cả các ánh xạ từ map cho trước trong tham số vào map hiện tại.
V remove(Object key)	Xóa ánh xạ của key được chỉ định nếu nó tồn tại trong map hiện tại.
int size()	Trả về số lượng cặp ánh xạ key-value có trong map.
NavigableMap<K, V> subMap(K fromKey, boolean from, K toKey, boolean to)	Trả về một NavigableMap chứa các phần tử có key nằm trong đoạn [from, to].
SortedMap<K, V> subMap(K from, K to)	Trả về một SortedMap chứa các phần tử nằm trong nửa khoảng [from, to).
SortedMap<K, V> tailMap(K from)	Trả về một SortedMap chứa các key lớn hơn hoặc bằng from.
NavigableMap<K, V> subMap(K from, boolean inclusive)	Trả về một phần của map hiện tại chứa các key lớn hơn from (hoặc bằng from nếu inclusive == true).
Collection<V> values()	Trả về một Collection các giá trị có trong map hiện thời.



# Nội dung tiếp theo

## Cây nhị phân cân bằng