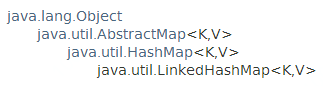
LinkedHashMap类的介绍

# 继承关系介绍

public class **LinkedHashMap<K,V>**

extends **HashMap**<K,V> implements Map<K,V>



# 功能介绍

A special constructor is provided to create **a linked hash map** whose order of iteration is the order in which its entries were **last accessed**, **from least-recently accessed to most-recently (access-order)**. This kind of map is well-suited to building **LRU caches**. Invoking the **put, putIfAbsent, get, getOrDefault, compute, computeIfAbsent, computeIfPresent, or merge methods** results in an access to the corresponding entry (assuming it exists after the invocation completes). The **replace** methods only result in an access of the entry if the value is replaced. The **putAll** method generates one entry access for each mapping in the specified map, in the order that key-value mappings are provided by the specified map's entry set iterator. No other methods generate entry accesses. In particular, operations on collection-views do not affect the order of iteration of the backing map.

# 构造方法

三个配置参数：

**int initialCapacity, float loadFactor, boolean accessOrder**

**accessOrder 指定存储模式**：添加顺序存储和访问顺序顺序，默认情况下是添加顺序存储，当指定accessOrder为true时，采用**访问顺序存储**。

LinkedHashMap()

Constructs **an empty insertion-ordered LinkedHashMap instance** with the default initial capacity (16) and load factor (0.75).

LinkedHashMap(int initialCapacity)

Constructs **an empty insertion-ordered LinkedHashMap instance** with the specified initial capacity and a default load factor (0.75).

LinkedHashMap(int initialCapacity, float loadFactor)

Constructs **an empty insertion-ordered LinkedHashMap instance** with the specified initial capacity and load factor.

LinkedHashMap(**int initialCapacity, float loadFactor, boolean accessOrder**)

Constructs an empty LinkedHashMap instance with the specified initial capacity, load factor and **ordering mode**.

LinkedHashMap(Map<? extends K,? extends V> m)

Constructs an **insertion-ordered LinkedHashMap** instance with the same mappings as the specified map.

# 方法介绍

## get方法

### V get(Object key)

Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.

### V getOrDefault(Object key, V defaultValue)

Returns the value to which the specified key is mapped, or **defaultValue** if this map contains no mapping for the key.

## void clear()

Removes all of the mappings from this map.

## boolean containsValue(Object value)

Returns true if this map maps one or more keys to the specified value.

## Set<Map.Entry<K,V>> entrySet()

Returns a Set view of the mappings contained in this map.

## void forEach(BiConsumer<? super K,? super V> action)

Performs the given action for each entry in this map until all entries have been processed or the action throws an exception.

## Set<K> keySet()

Returns a Set view of the keys contained in this map.

## protected boolean removeEldestEntry(Map.Entry<K,V> eldest)

Returns true if this map should remove its eldest entry.

## void replaceAll(BiFunction<? super K,? super V,? extends V> function)

Replaces each entry's value with the result of invoking the given function on that entry until all entries have been processed or the function throws an exception.

## Collection<V> values()

Returns a Collection view of the values contained in this map.