Map [Interface]: LinkedHashMap

Definition

LinkedHashMap is like HashMap with an additional feature that it maintains the insertion order. This linked list defines the iteration ordering, which is normally the order in which keys were inserted into the map. The insertion order is not affected if a key is re-inserted into the map.

This implementation spares its clients from unspecified ordering provided by Hashmap without incurring the increased cost associate with TreeMap.

A linked hash map has two parameters that affect its performance, the initial capacity and load factor. The penalty of choosing an high value for initial capacity is less severe than for HashMap as iteration times are unaffected by capacity.

Features of LinkedHashMap

- Contains values based on a key. It implements the Map interface and extends the HashMap class.
- It contains only unique elements
- It may have one null key and multiple null values
- It is non-synchronized.
- It is the same as HashMap with an additional feature that it maintains the insertion order.

Type parameters

- K – The type of keys maintained by this map.
- V – the type of mapped values in this map.

Each node of the LinkedHashMap is represented as:

- **Hash**: All the input keys are converted into a hash which is a shorter form of the key so that the search and insertion are faster.
- **Key**: Since this class extends HashMap, the data is stored in the form of a key-value pair. Therefore, this parameter is the key to the data.
- **Value**: For every key, there is a value associated with it. This parameter stores the value of the keys. Due to generics, this value can be of any form.
- **Next**: Since the LinkedHashMap stores the insertion order, this contains the address to the next node of the LinkedHashMap.
- **Previous**: This parameter contains the address to the previous node of the LinkedHashMap.

Constructor summary

LinkedHashMap()	Constructs an empty insertion-ordered linkedHashMap instance with the default initial capacity 16 and load factor 0.75.
LinedHashMap(int initalCapacity)	Creates an linkedhashMap that has an initial size specified by initialcapacity and the default load factor is 0.75.
LinkedHashMap(int initalCapacity, **float** loadfactor)	Creates a LinkedHashMap with initial capacity specified by initalCapacity and load factor specified by loadfactor.
LinkedHashMap(int initalCapacity, **float** loadfactor, **boolean** accesOrder)	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 same mapping as the specified map.

Methods summary

Modifier and Type	**Method and Description**
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 actions throws an exception.
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.
Set<**K**>	**keySet**() Returns a Set view of the keys contained in this map.
Collection<**V**>	**values**()

	Returns a Collection view of the values contained in this map.
	removeEldestEntry(Map.Entry<**K**,**V**> **eldest**) Returns true if this map should remove its eldest entry.

When is its use recommended?

- This Linked list is recommended in shopping cart where the items appear in order they where inserted.
- A LinkedHashMap is useful whenever you need the ordering of keys to match the ordering of insertion. For example, when you want to delete the oldest item.
-