

JAVA.UTIL.LINKEDHASHMAP.REMOVEELDESTENTRY METHOD

http://www.tutorialspoint.com/java/util/linkedhashmap_removeeldestentry.htm

Copyright © tutorialspoint.com

Description

The **java.util.LinkedHashMap.removeEldestEntry** method returns true if this map should remove its eldest entry. This method is invoked by put and putAll after inserting a new entry into the map. It provides the implementor with the opportunity to remove the eldest entry each time a new one is added. This is useful if the map represents a cache: it allows the map to reduce memory consumption by deleting stale entries.

Declaration

Following is the declaration for **java.util.LinkedHashMap.removeEldestEntry** method

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

Parameters

- **eldest** -- The least recently inserted entry in the map, or if this is an access-ordered map, the least recently accessed entry. This is the entry that will be removed if this method returns true. If the map was empty prior to the put or putAll invocation resulting in this invocation, this will be the entry that was just inserted; in other words, if the map contains a single entry, the eldest entry is also the newest.

Return Value

This method returns true if the eldest entry should be removed from the map; false if it should be retained.

Exception

- NA

Example

The following example shows the usage of java.util.LinkedHashMap.removeEldestEntry method.

```
package com.tutorialspoint;

import java.util.*;

public class LinkedHashMapDemo {

    private static final int MAX_ENTRIES = 5;

    public static void main(String[] args) {
        LinkedHashMap lhm = new LinkedHashMap(MAX_ENTRIES + 1, .75F, false) {

            protected boolean removeEldestEntry(Map.Entry eldest) {
                return size() > MAX_ENTRIES;
            }
        };

        lhm.put(0, "H");
        lhm.put(1, "E");
        lhm.put(2, "L");
        lhm.put(3, "L");
        lhm.put(4, "O");

        System.out.println("" + lhm);
    }
}
```

Let us compile and run the above program, this will produce the following result:

{0=H, 1=F, 2=L, 3=L, 4=0}

Loading [Mathjax]/jax/output/HTML-CSS/jax.js