IdentityHashMap class in Java

IdentityHashMap implements Map, Serializable and Clonable interfaces and extends AbstractMap class.
This class is not a general-purpose Map implementation. While this class implements the Map interface, it intentionally violates Map's general contract, which mandates the use of the equals method when comparing objects.

This class is used when the user requires the objects to be compared via reference.

Constructors

- **IdentityHashMap():** Constructs a new, empty identity hash map with a default expected maximum size.
- **IdentityHashMap(int expectedMaxSize):** Constructs a new, empty map with the specified expected maximum size.
- **IdentityHashMap(Map m):** Constructs a new identity hash map containing the keys-value mappings in the specified map.

Methods

- **clear()** – Removes all of the mappings from this map.
- **clone()** – Returns a shallow copy of this identity hash map: the keys and values themselves are not cloned.
- **containsKey(Object key)** – Returns true if this map contains a mapping for the specified key.
- **containsValue(Object value)** – Returns true if this map maps one or more keys to the specified value.
- **entrySet()** – Returns a Set view of the mappings contained in this map.
- **equals(Object o)** – Compares the specified object with this map for equality.
- **get(Object key)** – Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
- **hashCode()** – Returns the hash code value for this map.
- **isEmpty()** – Returns true if this map contains no key-value mappings.
- **keySet()** – Returns a Set view of the keys contained in this map.
- **put(K key, V value)** – Associates the specified value with the specified key in this map.
- **putAll(Map m)** – Copies all of the mappings from the specified map to this map.
- **remove(Object key)** – Removes the mapping for a key from this weak hash map if it is present.
- **size()** – Returns the number of key-value mappings in this map.
- **values()** – Returns a Collection view of the values contained in this map.

IdentityHashMap vs HashMap

- IdentityHashMap uses equality operator "==" for comparing keys and values while HashMap uses equals method for comparing keys and values inside Map.
- Since IdentityHashMap doesn't use equals() its comparatively faster than HashMap for object with expensive equals().
- IdentityHashMap doesn't require keys to be immutable as it is not relied on equals().

Implementation

```java
// Java code to demonstrate IdentityHashMap and
// illustration of how it is different from HashMap
import java.util.Map;
import java.util.HashMap;
import java.util.IdentityHashMap;

public class IdentityHashMapExample
{
    public static void main(String[] args)
    {
        // Creating HashMap and IdentityHashMap objects
        Map hm = new HashMap();
        Map ihm = new IdentityHashMap();

        // Putting key and value in HashMap and IdentityHashMap Object
        hm.put("hmkey","hmvalue");
        hm.put(new String("hmkey"),"hmvalue1");
        ihm.put("ihmkey","ihmvalue");
        ihm.put(new String("ihmkey"),"ihmvalue1");

        // Print Size of HashMap and WeakHashMap Object
        //hm.size() will print 1 since it compares the objects logically
        // and both the keys are same
        System.out.println("Size of HashMap--"+hm.size());

        //ihm.size() will print 2 since it compares the objects by reference
        System.out.println("Size of IdentityHashMap--"+ihm.size());


    }
}
```

Output:

```
Size of HashMap--1
Size of IdentityHashMap--2
```

This article is contributed by **Pratik Agarwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

IdentityHashMap put() Method in Java

IdentityHashMap get() Method in Java

IdentityHashMap hashCode() Method in Java

IdentityHashMap putAll() Method in Java

IdentityHashMap values() Method in Java

IdentityHashMap keySet() Method in Java

IdentityHashMap isEmpty() Method in Java

IdentityHashMap clear() Method in Java

IdentityHashMap entrySet() Method in Java

IdentityHashMap containsKey() Method in Java

IdentityHashMap size() Method in Java

IdentityHashMap remove() Method in Java

IdentityHashMap clone() Method in Java

IdentityHashMap equals() Method in Java

IdentityHashMap containsValue() Method in Java

Article Tags : Java Java-HashMap

Practice Tags : Java