

Stack Overflow is a community of 9.0 million programmers, just like you, helping each other. Join them; it only takes a minute:

[Sign up](#) 

[Home](#)

[PUBLIC](#)

 **Stack Overflow**

[Tags](#)

[Users](#)

[Jobs](#)

[TEAMS](#)

[+ Create Team](#)

HashMap with Null Key and Null Value

[Ask Question](#)

Consider the following Code :

```
import java.util.*;

class Employee {

    String name;

    public Employee(String nm) {
        this.name=nm;
    }
}

public class HashMapKeyNullValue {

    Employee e1;

    public void display(){

        Employee e2=null;
        Map map=new HashMap();

        map.put(e2, "25");
        System.out.println("Getting the Value When e2 is set as KEY");
        System.out.println("e2 : "+map.get(e2));
        System.out.println("e1 : "+map.get(e1));
        System.out.println("null : "+map.get(null));

        map.put(e1, "");
    }
}
```

This site uses cookies to deliver our services and to show you relevant ads and job listings. By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#). Your use of Stack Overflow's Products and Services, including the Stack Overflow Network, is subject to these policies and terms.

```

        map.put(null, null);    // null as key and null as value
        System.out.println("Getting the Value when setting null as KEY and null
as value");
        System.out.println("e2 : "+map.get(e2));
        System.out.println("e1 : "+map.get(e1));
        System.out.println("null : "+map.get(null));

        map.put(null, "30");
        System.out.println("Getting the Value when setting only null as KEY");
        System.out.println("e2 : "+map.get(e2));
        System.out.println("e1 : "+map.get(e1));
        System.out.println("null : "+map.get(null));
    }

    public static void main(String[] args) {

        new HashMapKeyNullValue().display();

    }
}

```

The Output of program is :

```

Getting the Value When e2 is set as KEY
e2 : 25
e1 : 25
null : 25
Getting the Value when e1 is set as KEY
e2 :
e1 :
null :
Getting the Value when setting null as KEY and null as value
e2 : null
e1 : null
null : null
Getting the Value when setting only null as KEY
e2 : 30
e1 : 30
null : 30

```

Here how e1, e2, and null as keys are related to each other. Is all three are assigned to same hashcode ? If yes. WHY ?

[java](#) [collections](#) [null](#) [hashmap](#) [hashcode](#)

asked Sep 19 '14 at 11:19



[Nizam](#)

68 1 1 6

The values of e1 and e2 are null. Java is pass-by-value, and will pass only the value of a variable, when that variable is used as call parameter. That is the variable will never be reassigned inside the function, and the parameter name in the function will have nothing to do with any variable. – [Joop Eggen](#) Sep 19 '14 at 11:25

possible duplicate of [HashMap having null as key](#) – [SimY4](#) Sep 19 '14 at 11:28

5 Answers

HashMap does not call hashCode when *null* is passed as key and null Key is handled as special case.

Put Method

HashMap puts *null* key in bucket 0 and maps *null* as key to passed value. it does it by linked list data structure it uses internally.

Linked list data structure used by *HashMap* (a static class in `HashMap.java`)

```
static class Entry<K,V> implements Map.Entry<K,V> {  
    final K key;  
    V value;  
    Entry<K,V> next;  
    final int hash;  
}
```

In Entry class the *K* is set to *null* and value mapped to value passed in put method.

Get Method

edited Jul 6 '17 at 8:08



GingerHead
1

answered Sep 19 '14 at 12:09



Sandeep Vaid
707 5 7

2 [PLUS ONE] For clear explanation and good exemplifying – [GingerHead](#) Jul 6 '17 at 8:12

Just fyi, The answer is perfectly correct, but there are slight changes about HashMap implementation since java 1.8. Instead of linked list a balanced trees can be used as "buckets" – [fascynacja](#) Jan 29 at 7:38

If you pass `null` as map key, it will go to `0` bucket . All values of null key will go there. That is why it returns same value, cause all keys you are providing are `null` and are in the same bucket of your HashMap.

answered Sep 19 '14 at 11:27



Kamil Klys
1,047 1 12 21

what if i inserted two values with NULL as key, v1 and v2. while fetching which one i will get???? Always last one ? – [dinesh kandpal](#) Aug 21 '17 at 6:10

@dineshkandpal yes, always the last one (v2) – [Alex Torson](#) Aug 31 '17 at 12:29

Incase of **null** key , Hashmap implementation consider it as special case and doesnot call hashCode method instead it stores Entry object to 0 bucket location.

A HashMap can only store one value per key. If you want to store more values, you have to use a MultivaluedHashMap (Google Guava and Apache Commons Collections contain implementations of such a map).

e1 and e2 have the value null, since you don't assign any object to them. So if you use those variables, the key of that map entry is also null, which leads to your result. Null doesn't have any hashCode, but is tolerated as key in the HashMap (there are other Map implementations which don't allow Null as key).

answered Sep 19 '14 at 11:23



[durni](#)

25.8k 5 64 76

When you put NULL to HashMap there is special check if you are trying to put NULL as key (called **putForNullKey()**). It is special case and works not like you are trying to put some object which is not null, and as you may see it even doesn't go to hash calculation.

```
public V put(K key, V value) {
    if (table == EMPTY_TABLE) {
        inflateTable(threshold);
    }
    if (key == null)
        return putForNullKey(value);
    int hash = hash(key);
    int i = indexFor(hash, table.length);
    for (Entry<K,V> e = table[i]; e != null; e = e.next) {
        Object k;
        if (e.hash == hash && ((k = e.key) == key || key.equals(k))) {
            V oldValue = e.value;
            e.value = value;
            e.recordAccess(this);
            return oldValue;
        }
    }

    modCount++;
    addEntry(hash, key, value, i);
}
```

```
        V oldValue = e.value;
        e.value = value;
        e.recordAccess(this);
        return oldValue;
    }
}
modCount++;
addEntry(0, null, value, 0);
return null;
}
```

answered Sep 19 '14 at 11:25



[Mykola Evpak](#)

146 6