Why does Hashtable not take null key?

Ask Question

Why does Hashtable not take a `null` key?

Also why does HashMap allow `null` keys?

What is the purpose of making these two classes Key behaviour so different?

java collections

edited Sep 26 '11 at 22:53
Bozho
461k 100 912 1036

asked Sep 26 '11 at 14:08
BOSS
1,540 4 23 47

1 Well, this might answer your question: stackoverflow.com/questions/40471/java-hashmap-vs-hashtable – Pieter Sep 26 '11 at 14:11

1 If you look into `AbstractMap` you will see here and there that NULL keys are specially handled. You can use null keys in `Hashtable` if you wrap them into objects (`NullKey`) and treat them specially. – dma_k Sep 26 '11 at 23:20

6 Answers

From the `Hashtable` JavaDoc:

In a nutshell, since `null` isn't an object, you can't call `.equals()` or `.hashCode()` on it, so the `Hashtable` can't compute a hash to use it as a key.

`HashMap` is newer, and has more advanced capabilities, which are basically just an improvement on the `Hashtable` functionality. As such, when `HashMap` was created, it was specifically designed to handle `null` values as keys and handles them as a special case.

Specifically, the use of `null` as a key is handled like this when issuing a `.get(key)` :

```
(key==null ? k==null : key.equals(k))
```

edited Sep 28 '11 at 12:48

answered Sep 26 '11 at 14:14

cdeszaq
21.9k 21 102 157

HashMap is not a wrapper around HashTable. It is a separate implementation. – EJP Sep 27 '11 at 0:12

@EJP - I didn't say that it actually *was* a wrapper around `Hashtable` , but rather that it wrapped the same *functionality*, since it does what `Hashtable` does and more. – cdeszaq Sep 27 '11 at 12:48

So when you said 'wrap' you didn't mean 'wrap'. Possibly you mean 'implement' or 'provide'. – EJP Sep 28 '11 at 10:13

@EJP - Yes, you're correct. "wrap" was a bad word to use. I've updated my answer to better reflect what I meant, as well as reality. – cdeszaq Sep 28 '11 at 12:49

It is just an implementation detail.

`Hashtable` is the older class, and its use is generally discouraged. Perhaps they saw the need for a null

I'm not sure I would call the handling of `null` values in a class's external interface an implementation detail, but I agree that it was likely a design oversight for `HashTable` that was corrected in `HashMap` . – cdeszaq Sep 26 '11 at 14:25

The `Map` interface says nothing of `null` values, so each implementation can choose :) – Bozho Sep 26 '11 at 14:27

Hashtable predates the collections framework, and was part of JDK 1.0. At that time null keys were probably considered not useful or not essential, and were thus forbidden. You might see it as a design error, just as the choice of the name `Hashtable` rather than `HashTable` .

Then, several years later, came the collections framework, and Hashtable was slightly modified to fit into the framework. But the behavior on null keys was not changed to keep backward compatibility.

Hashtable should be deprecated, IMHO.

answered Sep 26 '11 at 14:16

JB Nizet
504k 49 799 946

+1 for saying what I was going to say – Mansuro Sep 26 '11 at 14:34

I will let you know how the hashmap stores the objects internally:

HashMap stores the values through put(key,value) and gets the values thorugh `get(key)` . The process follows the concept of Hashing.

When we say `put(key,value)` - Internally `hashCode()` for the key is calculated and being taken as the input for `hashfunction()` to find the bucket location for storing.

In case of Collision - while calculating the `hashcode()` there may be a possibility the key is different but the `hashcode()` is same, at that time after finding out the bucket location the storing is done in the linked list. Please Note - While storing as the Map Entry both the key value is stored.

Regards, Anand

edited Apr 2 '13 at 9:14
Mariusz Jamro
18.5k 12 73 110

answered Oct 18 '12 at 7:58
Anand Builders
81 1 11

Apart from all the details given in other answers, this is how hashmap allow NULL Keys. If you look at the method `putForNullKey()` in Hashmap (JDK 5) , it reserves the index "0" for the null key. All values for the null key are kept inside the "0" index of the array.

There is nothing special about storing NULL value as all the put and lookup operations work based on the Key object.

In hashtable, Java does not have these mechanisms and hence hashtable does not support NULL key or values.

edited Apr 2 '13 at 9:14
Mariusz Jamro
18.5k 12 73 110

answered Oct 13 '11 at 14:32
java_mouse
1,214 3 13 29

They're two separate classes for two different things. Also, HashTable is synchronized. HashTable also came before HashMap, so naturally it would be less advanced. It probably didn't make sense to generate a null hashcode in early Java.

answered Sep 26 '11 at 15:05
Ian Macalinao
720 2 10 25
