GeeksforGeeks
A computer science portal for geeks

Custom Search

Practice GATE CS Placements Videos Contribute

Login/Register

HashMap Class Methods in Java with Examples | Set 1 (put(), get(), isEmpty() and size())

HashMap is a data structure that uses a hash function to map identifying values, known as keys, to their associated values. It contains “key-value” pairs and allows retrieving value by key.

The most impressive feature is it’s fast lookup of elements especially for large no. of elements. It is not synchronized by default but we can make it so by calling

```
Map myhash = Collections.synchronizedMap(hashMap);
```

at creation time, to prevent accidental unsynchronized access to the map.

These are various important hashmap class methods. This post explains: put(), get(), isEmpty() and size()

1. **put():** java.util.HashMap.put() plays role in associating the specified value with the specified key in this map. If the map previously contained a mapping for the key, the old value is replaced.
   Syntax:

   ```
   public V put(K key,V value)
   Parameters:
   key - key with which the specified value is to be associated
   value - value to be associated with the specified key
   Return: the previous value associated with
   key, or null if there was no mapping for key.
   ```

2. **get():** java.util.HashMap.get()method returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
   Syntax:

▲

```
public V get(Object key)
Parameters:
key - the key whose associated value is to be returned
Return: the value to which the specified
key is mapped, or null if this map contains no mapping for
the key.
```

3. **isEmpty():** java.util.HashMap.isEmpty() method returns true if the map contains no key-value mappings.

   Syntax:

   ```
   public boolean isEmpty()
   Return: true if this map contains no key-value mappings
   ```

4. **size():** java.util.HashMap.size() returns the number of key-value mappings in this map.

   Syntax:

   ```
   public int size()
   Return: the number of key-value mappings in this map.
   ```

Implementation to illustrate above methods

```java
// Java program illustrating use of HashMap methods -
// put(), get(), isEmpty() and size()
import java.util.*;
public class NewClass
{
    public static void main(String args[])
    {
        // Creation of HashMap
        HashMap<String, String> Geeks = new HashMap<>();

        // Adding values to HashMap as ("keys", "values")
        Geeks.put("Language", "Java");
        Geeks.put("Platform", "Geeks For geeks");
        Geeks.put("Code", "HashMap");
        Geeks.put("Learn", "More");

        System.out.println("Testing .isEmpty() method");

        // Checks whether the HashMap is empty or not
        // Not empty so printing the values
        if (!Geeks.isEmpty())
        {
            System.out.println("HashMap Geeks is notempty");

            // Accessing the contents of HashMap through Keys
            System.out.println("GEEKS : " + Geeks.get("Language"));
            System.out.println("GEEKS : " + Geeks.get("Platform"));
            System.out.println("GEEKS : " + Geeks.get("Code"));
            System.out.println("GEEKS : " + Geeks.get("Learn"));

            // size() method prints the size of HashMap.
            System.out.println("Size Of HashMap : " + Geeks.size());
        }
    }
}
```

Run on IDE

Output

```
Testing .isEmpty() method
HashMap Geeks is notempty
GEEKS : Java
GEEKS : Geeks For geeks
GEEKS : HashMap
GEEKS : More
Size Of HashMap : 4
```

What are the differences between HashMap and TreeMap?

1. HashMap implements Map interface while TreeMap implements SortedMap interface. A Sorted Map interface is a child of Map.
2. HashMap implements Hashing, while TreeMap implements Red-Black Tree(a Self Balancing Binary Search Tree). Therefore all differences between Hashing and Balanced Binary Search Tree apply here.
3. Both HashMap and TreeMap have their counterparts HashSet and TreeSet. HashSet and TreeSet implement Set interface. In HashSet and TreeSet, we have only key, no value, these are mainly used to see presence/absence in a set. For above problem, we can't use HashSet (or TreeSet) as we can't store counts. An example problem where we would prefer HashSet (or TreeSet) over HashMap (or TreeMap) is to print all distinct elements in an array.

Refer HashMap and TreeMap for details.

What are differences between HashMap and HashTable in Java?

- HashMap is non synchronized. It is not-thread safe and can't be shared between many threads without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads.
- HashMap allows one null key and multiple null values whereas Hashtable doesn't allow any null key or value.
- HashMap is generally preferred over HashTable if thread synchronization is not needed
- HashMap is an advanced version and improvement on the Hashtable. HashMap was created later.

Please Refer Differences between HashMap and HashTable in Java for details.

What are differences between HashMap and HashSet?

- HashMap stores key value pairs (for example records of students as value and roll number as key) and HashSet stores only keys (for example a set if integers).
- HashSet internally uses HashMap to store keys

Please refer HashSet in Java for details.

Hashmap methods in Java with Examples | Set 2 (keySet(), values(), containsKey())

Reference:
https://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html

This article is contributed by **Mohit Gupta**. 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.

GATE CS Corner **Company Wise Coding Practice**

Java Java-Collections

Recommended Posts:

Hashmap methods in Java with Examples | Set 2 (keySet(), values(), containsKey()..)
Java.util.HashMap in Java
LinkedHashMap class in Java with Example
HashMap and TreeMap in Java
Vector vs ArrayList in Java
Java.lang.String.compareTo()
Templates in C++ vs Generics in Java
Converting Text to Speech in Java
Difference between x++ and x=x+1 in Java
JavaBean class in Java

(Login to Rate and Mark)

1.6 Average Difficulty : **1.6/5.0**
Based on **6** vote(s)

Add to TODO List

Mark as DONE

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments Share this post! ▲