Java Hashtable class

Java Hashtable class implements a hashtable, which maps keys to values. It inherits Dictionary class and implements the Map interface.

The important points about Java Hashtable class are:

- A Hashtable is an array of list. Each list is known as a bucket. The position of bucket is identified by calling the hashcode() method. A Hashtable contains values based on the key.
- It contains only unique elements.
- It may have not have any null key or value.
- It is synchronized.

Hashtable class declaration

Let's see the declaration for java.util.Hashtable class.

```
public class Hashtable<K,V> extends Dictionary<K,V> implements Map<K,V>, Cloneable, Serializable
```

Hashtable class Parameters

Let's see the Parameters for java.util.Hashtable class.

- **K**: It is the type of keys maintained by this map.
- **V**: It is the type of mapped values.

Constructors of Java Hashtable class

Constructor	Description
Hashtable()	It is the default constructor of hash table it instantiates the Hashtable class.
Hashtable(int size)	It is used to accept an integer parameter and creates a hash table that has an initial size specified by integer value size.
Hashtable(int size, float fillRatio)	It is used to create a hash table that has an initial size specified by size and a fill ratio specified by fillRatio.

Methods of Java Hashtable class

Method	Description
void clear()	It is used to reset the hash table.
boolean contains(Object value)	This method return true if some value equal to the value exist within the hash table, else return false.
boolean containsValue(Object value)	This method return true if some value equal to the value exists within the hash table, else return false.
boolean containsKey(Object key)	This method return true if some key equal to the key exists within the hash table, else return false.
boolean isEmpty()	This method return true if the hash table is empty; returns false if it contains at least one key.
void rehash()	It is used to increase the size of the hash table and rehashes all of its keys.
Object get(Object key)	This method return the object that contains the value associated with the key.
Object remove(Object key)	It is used to remove the key and its value. This method return the value associated with the key.
int size()	This method return the number of entries in the hash table.

Java Hashtable Example

```
import java.util.*;
class TestCollection16{
 public static void main(String args[]){
  Hashtable<Integer,String> hm=new Hashtable<Integer,String>();

  hm.put(100,"Amit");
  hm.put(102,"Ravi");
  hm.put(101,"Vijay");
  hm.put(103,"Rahul");

  for(Map.Entry m:hm.entrySet()){
   System.out.println(m.getKey()+" "+m.getValue());
  }
 }
```

```
}
```

Test it Now

Output:

```
103 Rahul
102 Ravi
101 Vijay
100 Amit
```

Java Hashtable Example: remove()

```
import java.util.*;
public class HashtableExample {
  public static void main(String args[]) {
  // create and populate hash table
  Hashtable<Integer, String> map = new Hashtable<Integer, String>();
  map.put(102,"Let us C");
  map.put(103, "Operating System");
  map.put(101, "Data Communication and Networking");
  System.out.println("Values before remove: "+ map);
  // Remove value for key 102
  map.remove(102);
  System.out.println("Values after remove: "+ map);
  }
}
```

Output:

```
Values before remove: {103=Operating System, 102=Let us C, 101=Data Communication and Networking}
Values after remove: {103=Operating System, 101=Data Communication and Networking}
```

Java Hashtable Example: Book

```
import java.util.*;
class Book {
int id;
```

```
String name,author,publisher;
int quantity;
public Book(int id, String name, String author, String publisher, int quantity) {
    this.id = id;
    this.name = name;
    this.author = author;
    this.publisher = publisher;
    this.quantity = quantity;
}
}
public class HashtableExample {
public static void main(String[] args) {
    //Creating map of Books
    Map<Integer,Book> map=new Hashtable<Integer,Book>();
    //Creating Books
    Book b1=new Book(101,"Let us C","Yashwant Kanetkar","BPB",8);
    Book b2=new Book(102,"Data Communications & Networking","Forouzan","Mc Graw Hill",4);
    Book b3=new Book(103,"Operating System","Galvin","Wiley",6);
    //Adding Books to map
    map.put(1,b1);
    map.put(2,b2);
    map.put(3,b3);
    //Traversing map
    for(Map.Entry<Integer, Book> entry:map.entrySet()){
        int key=entry.getKey();
        Book b=entry.getValue();
        System.out.println(key+" Details:");
        System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);
    }
}
}
```

Output:

```
3 Details:
103 Operating System Galvin Wiley 6
2 Details:
102 Data Communications & Networking Forouzan Mc Graw Hill 4
```

```
1 Details:
101 Let us C Yashwant Kanetkar BPB 8
```

← prev next →

Share this page

Latest 4 Tutorials