**hash table**

1. basic **hash table** operation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows basic operations on Hashtable like creating hashtable object, adding key-value pair, getting the value based on key, checking hashtable is empty or not, removing an element, and size of the hashtable. |  |  | | --- | | **Code:** | | [?](https://www.java2novice.com/java-collections-and-util/hashtable/basic-operations/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | package com.javacoffee.hashtable;    import java.util.Hashtable;    public class MyHashtableOperations {        public static void main(String a[]){          //Create hashtable instance          Hashtable<String,String> ht = new Hashtable<String,String>();          //add key-value pair to hashtable          ht.put("first", "FIRST INSERTED");          ht.put("second", "SECOND INSERTED");          ht.put("third","THIRD INSERTED");          System.out.println(ht);          //getting value for the given key from hashtable          System.out.println("Value of key 'second': "+ht.get("second"));          System.out.println("Is Hashtable empty? "+ht.isEmpty());          ht.remove("third");          System.out.println(ht);          System.out.println("Size of the Hashtable: "+ht.size());      }  } | | |
| |  | | --- | | **Output:** | | {third=THIRD INSERTED, second=SECOND INSERTED, first=FIRST INSERTED}  Value of key 'second': SECOND INSERTED  Is Hashtable empty? false  {second=SECOND INSERTED, first=FIRST INSERTED}  Size of the Hashtable: 2 | |

1. how to iterate through hash table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows how to read elements from Hashtable. You can iterate through each and every element by getting all keys as set object. Using each element as a key from set, you can values from Hashtable. |  |  | | --- | | **Code:** | | [?](https://www.java2novice.com/java-collections-and-util/hashtable/iterate/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | package com.javacoffee.hashtable;    import java.util.Hashtable;  import java.util.Set;    public class MyHashtableRead {        public static void main(String a[]){            Hashtable<String, String> hm = new Hashtable<String, String>();          //add key-value pair to Hashtable          hm.put("first", "FIRST INSERTED");          hm.put("second", "SECOND INSERTED");          hm.put("third","THIRD INSERTED");          System.out.println(hm);          Set<String> keys = hm.keySet();          for(String key: keys){              System.out.println("Value of "+key+" is: "+hm.get(key));          }      }  } | | |
| |  | | --- | | **Output:** | | {third=THIRD INSERTED, second=SECOND INSERTED, first=FIRST INSERTED}  Value of third is: THIRD INSERTED  Value of second is: SECOND INSERTED  Value of first is: FIRST INSERTED | |

1. **How to search a key in Hashtable?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows how to search a key from the Hashtable. The method containsKey() helps us to find whether the specified key exists or not. |  |  | | --- | | **Code:** | | [?](https://www.java2novice.com/java-collections-and-util/hashtable/key-search/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | package com.javacoffee.hashtable;    import java.util.Hashtable;    public class MyHashtableKeySearch {        public static void main(String a[]){          Hashtable<String, String> hm = new Hashtable<String, String>();          //add key-value pair to Hashtable          hm.put("first", "FIRST INSERTED");          hm.put("second", "SECOND INSERTED");          hm.put("third","THIRD INSERTED");          System.out.println(hm);          if(hm.containsKey("first")){              System.out.println("The Hashtable contains key first");          } else {              System.out.println("The Hashtable does not contains key first");          }          if(hm.containsKey("fifth")){              System.out.println("The Hashtable contains key fifth");          } else {              System.out.println("The Hashtable does not contains key fifth");          }      }  } | | |
| |  | | --- | | **Output:** | | {third=THIRD INSERTED, second=SECOND INSERTED, first=FIRST INSERTED}  The Hashtable contains key first  The Hashtable does not contains key fifth | |

1. how to search value in hash table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows how to search a value from the Hashtable. The method containsValue() helps us to find whether the specified value exists or not. |  |  | | --- | | **Code:** | | [?](https://www.java2novice.com/java-collections-and-util/hashtable/value-search/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | package com.javacoffee.hashtable;    import java.util.Hashtable;    public class MyHashtableValueSearch {        public static void main(String a[]){          Hashtable<String, String> hm = new Hashtable<String, String>();          //add key-value pair to Hashtable          hm.put("first", "FIRST INSERTED");          hm.put("second", "SECOND INSERTED");          hm.put("third","THIRD INSERTED");          System.out.println(hm);          if(hm.containsValue("SECOND INSERTED")){              System.out.println("The Hashtable contains value SECOND INSERTED");          } else {              System.out.println("The Hashtable does not contains value SECOND INSERTED");          }          if(hm.containsValue("first")){              System.out.println("The Hashtable contains value first");          } else {              System.out.println("The Hashtable does not contains value first");          }      }  } | | |
| |  | | --- | | **Output:** | | {third=THIRD INSERTED, second=SECOND INSERTED, first=FIRST INSERTED}  The Hashtable contains value SECOND INSERTED  The Hashtable does not contains value first | |

1. how to delete all element in hash table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows how to delete all key-values at one call from Hashtable. By calling clear() method, we can remove all elements from Hashtable at once. |  |  | | --- | | **Code:** | | package com.javacoffee.hashtable;  import java.util.Hashtable;  public class MyHashtableClear {  public static void main(String a[]){  Hashtable<String, String> hm = new Hashtable<String, String>();  //add key-value pair to Hashtable  hm.put("first", "FIRST INSERTED");  hm.put("second", "SECOND INSERTED");  hm.put("third","THIRD INSERTED");  System.out.println("My Hashtable content:");  System.out.println(hm);  System.out.println("Clearing Hashtable:");  hm.clear();  System.out.println("Content After clear:");  System.out.println(hm);  }  } | |
| |  | | --- | | **Output:** | | My Hashtable content:  {third=THIRD INSERTED, second=SECOND INSERTED, first=FIRST INSERTED}  Clearing Hashtable:  Content After clear:  {} | |

Hashset

1. basic hashset operation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows basic operations on HashSet object like creating object, adding elements, verifying whether the hashset is empty or not, removing an element, size of the hashset, and to check whether an object exists or not. |  |  | | --- | | **Code:** | | [?](https://www.java2novice.com/java-collections-and-util/hashset/basic-operations/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | package com.javacoffee.hashset;    import java.util.HashSet;    public class MyBasicHashSet {        public static void main(String a[]){          HashSet<String> hs = new HashSet<String>();          //add elements to HashSet          hs.add("first");          hs.add("second");          hs.add("third");          System.out.println(hs);          System.out.println("Is HashSet empty? "+hs.isEmpty());          hs.remove("third");          System.out.println(hs);          System.out.println("Size of the HashSet: "+hs.size());          System.out.println("Does HashSet contains first element? "+hs.contains("first"));      }  } | | |
| |  | | --- | | **Output:** | | [second, third, first]  Is HashSet empty? false  [second, first]  Size of the HashSet: 2  Does HashSet contains first element? true | |

2.how to iterate hashset.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows how to read all elements from the HashSet objects. You can iterate through HashSet by getting Iterator object. By calling iterator() method, you can get Iterator object. |  |  | | --- | | **Code:** | | [?](http://www.java2novice.com/java-collections-and-util/hashset/iterate/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | package com.javacoffee.hashset;    import java.util.HashSet;  import java.util.Iterator;    public class MyHashSetRead {        public static void main(String a[]){          HashSet<String> hs = new HashSet<String>();          //add elements to HashSet          hs.add("first");          hs.add("second");          hs.add("third");          Iterator<String> itr = hs.iterator();          while(itr.hasNext()){              System.out.println(itr.next());          }      }  } | | |
| |  | | --- | | **Output:** | | second  third  first | |
| |  | | --- | |  | |

1. how to delete all element in hashset.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Description:** | | Below example shows how to delete all entries at one call from HashSet. By calling clear() method, we can remove all elements from HashSet at once. |  |  | | --- | | **Code:** | | [?](https://www.java2novice.com/java-collections-and-util/hashset/delete-all/)   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | package com.javacoffee.hashset;    import java.util.HashSet;    public class MyHashSetClear {        public static void main(String a[]){          HashSet<String> hs = new HashSet<String>();          //add elements to HashSet          hs.add("first");          hs.add("second");          hs.add("third");          System.out.println("My HashSet content:");          System.out.println(hs);          System.out.println("Clearing HashSet:");          hs.clear();          System.out.println("Content After clear:");          System.out.println(hs);      }  } | | |
| |  | | --- | | **Output:** | | My HashSet content:  [second, third, first]  Clearing HashSet:  Content After clear:  [] | |