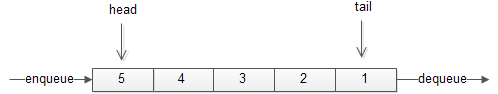
Hash set

* Duplicates Elements are not allowed
* If you add duplicates also –After Compiling – Duplicates will be not printed in the Console
* Insertion Order is Not Preserved – Generally insertion is not preferred. if you want to for insertion you may go

#### Elements arranged in Hash set are



## Methods in hash set

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Description | Methods | Notes |
| 1 | Difference between two sets | set1.removeall(set2) | Modifies set1 to remove elements present in set2. |
| 2 | Unique Elements in two sets or Union of two sets | set1.Addall(set2) | Modifies set1 to add all elements from set2. |
| 3 | Common elements in two sets  Or intersection of two sets | set1.retainall(set2) | Modifies set1 to retain only elements present in both set1 and set2. |

#### Note: -- hash set don’t allow the Duplicate elements

# Linked-Hash set

*Methods, which are available in Hash set, are available in Liked List as well*

Here we will some differences

|  |  |  |
| --- | --- | --- |
| S.No. | Hash Set | Linked Hash set |
| 1 | Duplicates are not allowed | Duplicates are not allowed |
| 2 | Insertion Order not Preserved | Insertion order is Preserved |
| 3 | Data Structure – Hash Table | Data Structure – Hash Table + Linked List |
| 4 | Elements are printed in Random Order in Console | Elements are printed in Given Order- Here we have Concept of Linked List –in Hash set |
| 5 | Given – 1,2,3,4  Console --- 4,1,3,2 | Given – 1,2,3,4  Console --- 1,2,3,4 |
| 6 | Initial Capacity = 16 and load factor = 0.25 | Initial Capacity = 16 and load factor = 0.75 |

# Queue (Interface) ------Deque (interface)

|  |  |  |  |
| --- | --- | --- | --- |
| S.no. | Description | Method | When elements are Empty –what its Returns |
| 1 | To add the elements into Que | add () | Exception |
| 2 | To add the elements into Que | offer () | False |
| 3 | To Return the elements into Que | element () | Exception |
| 4 | To Return the elements into Que | peek () | Null |
| 5 | To Return and Remove the elements into Que | remove () | Exception |
| 6 | To Return and Remove the elements into Que | poll () | Null |

# Map (Interface)

1. Map is independent
2. No relation between Map and Collection interface
3. When –when you want to represent Objects in the form of Key and Value – pairs

|  |  |  |
| --- | --- | --- |
| S.no | Key | Value |
| 1 | 101 | Lion |
| 2 | 102 | Cheetah |
| 3 | 103 | Tiger |
| 4 | 104 | Rider |
| 5 | Null | Lion |
| 6 | 105 | Null |
| 7 | 106 | Null |

1. Keys should be Unique
2. Value may be duplicated
3. Key + value = pair
4. Combination Single key + Single value = One Entry
5. So, Hash map is a Collection of Entries

### Similarities of Hash Map and Hash Table

1. When we want to work with Key and Value pair we will go with Hash Map or Hash Table concept
2. Under lying Data Structure for both the classes is Hash Table , Internally It will follow-Hash code- to store the Data

|  |  |  |
| --- | --- | --- |
| **S.no** | **Hash-Map** | **Hash Table** |
| 1 | **Non - Synchronized -**  1. Multiple threads can work on same method at a time - No waiting Period | **Synchronized:**  1. When we have n number of methods in the Hash Table, multiple threads cannot work on the same method simultaneously. 2. Threads have to work one after the other. 3. Other threads should wait while one thread is working. 4. Only one thread is allowed to execute the method at a time. |
| 2 | Not Thread Safe | Thread Safe |
| 3 | Performance is faster | Performance is Poor |
| 4 | 1. For A key It will accept Single Null --- Values May be Multiple Nulls  2. Nulls are accepted | Nulls Cannot accepted Either as Key or Value |

### Difference between Hash Map and Hash Table

## Classes in Map

## 1. Hash map (class)

1. In hash map Under- Lying Data Structure is Hash-table
2. Insertion order not preserved. - Elements we have stored in the given order will not be stored in the same order.
3. Keys should be Unique
4. Duplicate Keys are Not allowed
5. Duplicate values are Not allowed
6. Only Null key is allowed
7. Duplicate null values may be allowed
8. Searching will be faster in Hash Map

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods in Hash map Class** | | | |
| ***S.no.*** | ***Description*** | ***Methods*** | ***Notes*** |
| 1 | To add a pair in Hash map | put (key, value) |  |
| 2 | To add a Hash map collection | putAll(map collection) |  |
| 3 | To get the Value from the key | get(key) | which will return value based on the key we have passed |
| 4 | To remove key and value | remove(key) | which will remove entire entry key and value |
| 5 | To know key | contains key(key) | 1.key present - true  2..key not present-false |
| 6 | To know value | contains value(value) | 1.Value present - true  2.Value not present-false |
| 7 | To check Hash map is Empty | is Empty() | 1. Data is Present returns True 2. Data is not Present returns False |
| 8 | To know How many entries in the Collection | size () |  |
| 9 | To clear all the Entries from the Hash map | clear () |  |
| 10 | To return keys | Keyset () | 1. This method will get all keys in as et – return as **Object** |
| 11 | To return Values | Values () | This will return all the Values as a **Collection** |
| 12 | It will return all Entry from the Hash Map- as a Set individually | entrySet () |  |

1. Special Methods are keyset () and values ()

|  |
| --- |
| Keyset () --------Object -------------Unique --------As Keys are Unique |
| Values () --------Collection ----------------Duplicate –Values -----As Values Duplicates are Allowed |

## 2. Entry interface

1. Hash-Map has n number of Entries – we can represent these entries by Entry interface
2. Entry interface certain methods
3. We can use those methods only on the Entry objects from the Hash map

#### Methods in Entry interface

1. getkey()

|  |  |  |
| --- | --- | --- |
|  | Key | Value |
| e | 101 | x |

1. when e is representing key and value (Single object)– when we write **e. getkey()---** it will get the key

Result is -----101

1. when e is representing key and value – when we write e. **getvalue()**--- it will get the value

Result is -----x

1. when e is representing key and value – when we write e. **setvalue(object )**--- it will replace the value

Result is -----Object

1. To print Key and Value line by line we will go with For loop

## 3. Hash table (class)

1. Default capacity of Hash-table is 11
2. Load factor is 0.75
3. Declaration

Example of Hash Table

<https://gist.github.com/Sameer-Programmer/b2494768f8bf9c9112f3f4b3fa31cbfa>

Programs Related to Hash Set, Hash-Map, Has Table

<https://github.com/Sameer-Programmer/Java/tree/master/example/Sameer_Collection_Class>

# [HashTable(classes)](https://www.mindmeister.com/app/map/2871857060?t=1ox7ICkZXn)