

Map Interface Methods:-

Shambhu Kumar



[@Kumarsam07](#)



[@javac_java](#)



<http://linkedin.com/in/kumarsam07>

1. Map Interface Methods:

The **Map** interface in Java is part of the **java.util** package and represents a collection of key-value pairs, where each key is unique. It provides methods for adding, removing, and accessing elements based on keys.

Here is a detailed list of all the methods in the **Map** interface, along with their descriptions:

1. Basic Operations

1. **V put(K key, V value)**

- **Description:** Associates the specified value with the specified key in the map.
- **Returns:** The previous value associated with the key, or null if there was no mapping.
- **Throws:** NullPointerException if the key or value is null (in some implementations).

2. **void putAll(Map<? extends K, ? extends V> m)**

- **Description:** Copies all key-value pairs from the specified map to this map.
- **Throws:** NullPointerException if the specified map is null.

3. V get(Object key)

- **Description:** Returns the value associated with the specified key, or null if the map contains no mapping for the key.

4. V remove(Object key)

- **Description:** Removes the mapping for the specified key if it exists.
- **Returns:** The value previously associated with the key, or null if no mapping was found.

5. boolean containsKey(Object key)

- **Description:** Checks if the map contains a mapping for the specified key.
- **Returns:** true if the key exists, otherwise false.

6. boolean containsValue(Object value)

- **Description:** Checks if the map contains one or more keys mapped to the specified value.
- **Returns:** true if the value exists, otherwise false.

7. void clear()

- **Description:** Removes all mappings from the map.

8. int size()

- **Description:** Returns the number of key-value mappings in the map.

9. boolean isEmpty()

- **Description:** Checks if the map contains no key-value mappings.
- **Returns:** true if the map is empty, otherwise false.

2. Bulk Retrieval

10. **Set<K> keySet()**

- **Description:** Returns a set view of the keys contained in the map.

11. **Collection<V> values()**

- **Description:** Returns a collection view of the values contained in the map.

12. **Set<Map.Entry<K, V>> entrySet()**

- **Description:** Returns a set view of the key-value mappings contained in the map. Each entry is represented as a Map.Entry object.

3. Methods Added in Java 8

13. **V getOrDefault(Object key, V defaultValue)**

- **Description:** Returns the value associated with the specified key, or the provided defaultValue if the key is not found in the map.

14. **V putIfAbsent(K key, V value)**

- **Description:** Associates the specified value with the specified key only if the key is not already mapped to a value.
- **Returns:** The current value associated with the key, or null if no mapping existed.

15. **boolean remove(Object key, Object value)**

- **Description:** Removes the entry for the specified key only if it is currently mapped to the specified value.

- **Returns:** true if the entry was removed, otherwise false.
 - 16. **boolean replace(K key, V oldValue, V newValue)**
 - **Description:** Replaces the value for the specified key only if it is currently mapped to the specified old value.
 - **Returns:** true if the value was replaced, otherwise false.
 - 17. **V replace(K key, V value)**
 - **Description:** Replaces the value for the specified key only if it is already mapped to some value.
 - **Returns:** The previous value associated with the key, or null if no mapping existed.
 - 18. **void forEach(BiConsumer<? super K, ? super V> action)**
 - **Description:** Performs the given action for each entry in the map until all entries have been processed or the action throws an exception.
 - 19. **void replaceAll(BiFunction<? super K, ? super V, ? extends V> function)**
 - **Description:** Replaces each entry's value with the result of applying the specified function to that entry.
-

2. Map.Entry Interface Methods:

The **Map.Entry** interface in Java is a nested interface inside the **Map** interface, representing a key-value pair (entry) in a map. It provides methods to access and manipulate individual entries in a map. This interface is part of the **java.util** package.

Here is a comprehensive list of all the methods in the **Map.Entry** interface, along with their descriptions:

1. Methods for Accessing Key-Value Pairs

1. K getKey()

- **Description:** Returns the key corresponding to this entry in the map.
- **Usage:** Used to retrieve the key of the current entry.

2. V getValue()

- **Description:** Returns the value associated with this entry in the map.
- **Usage:** Used to retrieve the value of the current entry.

2. Methods for Updating Values

3. V setValue(V value)

- **Description:** Replaces the value corresponding to this entry with the specified value.
- **Returns:** The previous value associated with the entry.
- **Throws:**
 - UnsupportedOperationException if the operation is not supported by the map.
 - ClassCastException if the specified value's type is not compatible.
 - NullPointerException if the specified value is null and the map does not permit null values.
 - IllegalArgumentException if some property of the specified value prevents it from being stored in the map

=====END=====