

# Sets and Maps

By Ian Ludden, Tyler Whitehouse, and Luke Zhang

## Sources

[Big Java](#), Fifth Edition (Horstmann); Sections 16.1 and 16.2

Oracle.com – The Java Tutorials: The Set Interface, The Map Interface, Set Implementations, Map Implementations

## Summary

Definitions:

In general, a set is a collection of items. A mathematical set is a group of elements that does not have duplicates (e.g., the set of natural numbers: {1, 2, 3, ...}). The Set interface in Java is a subtype of the Collection interface with no new methods; the only difference is that items cannot appear twice.

A map is a type of data structure that stores values by assigning keys that correspond to them. It is similar to a mathematical function: just as one x-value can point to only one y-value in a mathematical function, each key in a map can only point to one value. The keys and values of a Map object can be of different object types. The Map interface in Java is found in the java.util package and has multiple subtypes and subclasses, such as ConcurrentMap<K, V>, NavigableMap<K, V>, and Hashtable. Its notable methods include: get(Object key), which returns the value the given key maps to or null if the key is not in the map; put(K key, V value), which adds the given key/value pair to the map; remove(Object key), which removes the given key and its value from the map; and the containsKey(Object key) and containsValue(Object value) methods, which return true if the map contains the given key or value and false otherwise.

Uses:

Sets are useful for finding and eliminating duplicates in a given list, because if you try to add a duplicate item, the add () method will return false and the item will not be added. Its most commonly used implementations include the HashSet, TreeSet, and LinkedHashSet classes. HashSet does not order its elements, but it only takes constant-time for its add, remove, and contains methods. On the other hand, TreeSet orders its elements based on their natural ordering (e.g., alphabetic order for String objects). The LinkedHashSet is a sort of balance between HashSet and TreeSet: it runs almost as quickly as HashSet, but orders items based on the order they are inserted.

Maps are useful for a variety of tasks; some examples of key/value pairs include item/price, ID number/user, and word/frequency in document. As with Sets, there are three common Map implementations: HashMap, TreeMap, and LinkedHashMap. These classes are analogous to their corresponding set classes in terms of runtime and ordering.

Example declaration and initialization code:

```
TreeSet<String> uniqueWordsAlphabetized = new TreeSet<String>();  
HashMap<String, Integer> usersByID = new HashMap<String, Integer>();
```

In the above examples, if <String> or <String, Integer> was left off, each element in the set would be of type Object and the key and value types for the map would be of type Object.