Map

By Rahul Barve

Map

- It stores elements in the form of key-value pairs.
- For every key, there is a value associated.

`Map`

- The key has to be unique, but values may be duplicates.
- Hence, the key specific class must override `hashCode()` and `equals()`.

Map

- It has several Implementations:
  - `Hashtable`
  - `HashMap`
  - `Properties`

Sorted Collections

Sorted Collections

- Whenever a collection is holding multiple data values, it is a very common requirement to perform sorting on those values.

Sorted Collections

- Collections Framework provides 2 APIs to perform sorting.
- `SortedSet` **and** `SortedMap`

SortedSet

- It is an extension to `Set`.
- Does not permit duplicate values, but stores elements in a sorted order.
- It is implemented by `TreeSet`.

SortedMap

- It is an extension to `Map`.
- Stores elements in the form of key-value pairs.

SortedMap

- Performs sorting on the basis of keys.
- It is implemented by `TreeMap`.

Sorting Customization

Sorting Customization

- Collection Framework also provides a mechanism to customize the sorting algorithm.
- It is especially required in case of user defined objects.

Sorting Customization

- To customize the sorting algorithm, there are 2 APIs provided:
  - `java.lang.Comparable`
  - `Java.util.Comparator`

Comparable

- Implemented by a class of which objects are to be sorted.
- Useful to provide a default sorting algorithm.

Comparator

- Implemented by a class that provides the sorting algorithm.
- Used to customize the sorting algorithm.