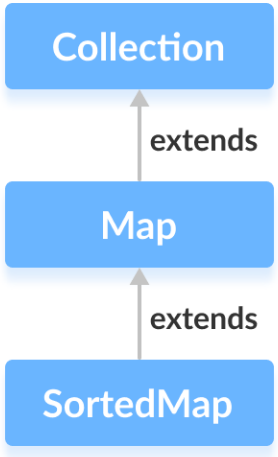
Java SortedMap Interface

In this tutorial, we will learn about the Java SortedMap interface and its methods.

The `SortedMap` interface of the Java collections framework provides sorting of keys stored in a map.

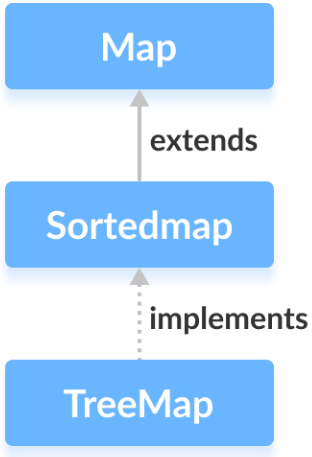
It extends the Map interface.



Class that implements SortedMap

Since `SortedMap` is an interface, we cannot create objects from it.

In order to use the functionalities of the `SortedMap` interface, we need to use the class `TreeMap` that implements it.



How to use SortedMap?

To use the `SortedMap`, we must import the `java.util.SortedMap` package first. Once we import the package, here's how we can create a sorted map.

```
// SortedMap implementation by TreeMap class
SortedMap<Key, Value> numbers = new TreeMap<>();
```

We have created a sorted map called `numbers` using the `TreeMap` class.

Here,

- `Key` - a unique identifier used to associate each element (value) in a map
- `Value` - elements associated by keys in a map

Here, we have used no arguments to create a sorted map. Hence the map will be sorted naturally (ascending order).

Methods of SortedMap

The `SortedMap` interface includes all the methods of the `Map` interface. It is because `Map` is a super interface of `SortedMap`.

Besides all those methods, here are the methods specific to the `SortedMap` interface.

- **comparator()** - returns a comparator that can be used to order keys in a map
- **firstKey()** - returns the first key of the sorted map
- **lastKey()** - returns the last key of the sorted map

- **headMap(key)** - returns all the entries of a map whose keys are less than the specified `key`
- **tailMap(key)** - returns all the entries of a map whose keys are greater than or equal to the specified `key`
- **subMap(key1, key2)** - returns all the entries of a map whose keys lies in between `key1` and `key2` including `key1`

To learn more, visit Java SortedMap (official Java documentation).

Implementation of SortedMap in TreeMap Class

```java
import java.util.SortedMap;
import java.util.TreeMap;

class Main {

    public static void main(String[] args) {
        // Creating SortedMap using TreeMap
        SortedMap<String, Integer> numbers = new TreeMap<>();

        // Insert elements to map
        numbers.put("Two", 2);
        numbers.put("One", 1);
        System.out.println("SortedMap: " + numbers);


        // Access the first key of the map
        System.out.println("First Key: " + numbers.firstKey());

        // Access the last key of the map
        System.out.println("Last Key: " + numbers.lastKey());

        // Remove elements from the map
        int value = numbers.remove("One");
        System.out.println("Removed Value: " + value);
    }
}
```

Output

```
SortedMap: {One=1, Two=2}
First Key: One
Last Key: Two
Removed Value: 1
```