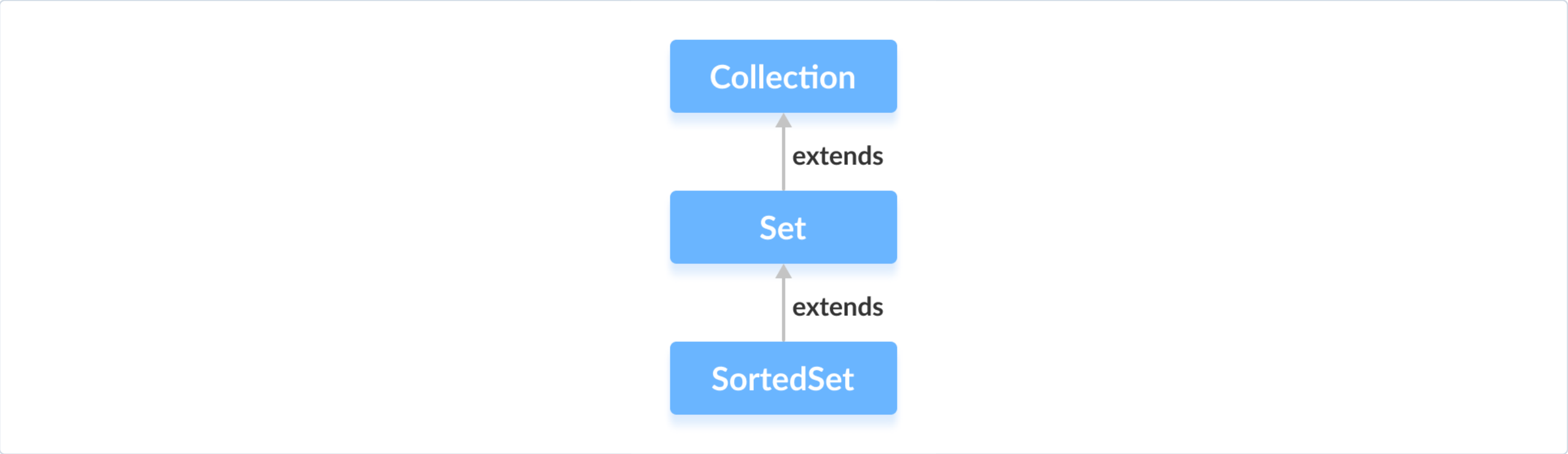
Java SortedSet Interface

In this tutorial, we will learn about the SortedSet interface in Java and its methods with the help of an example.

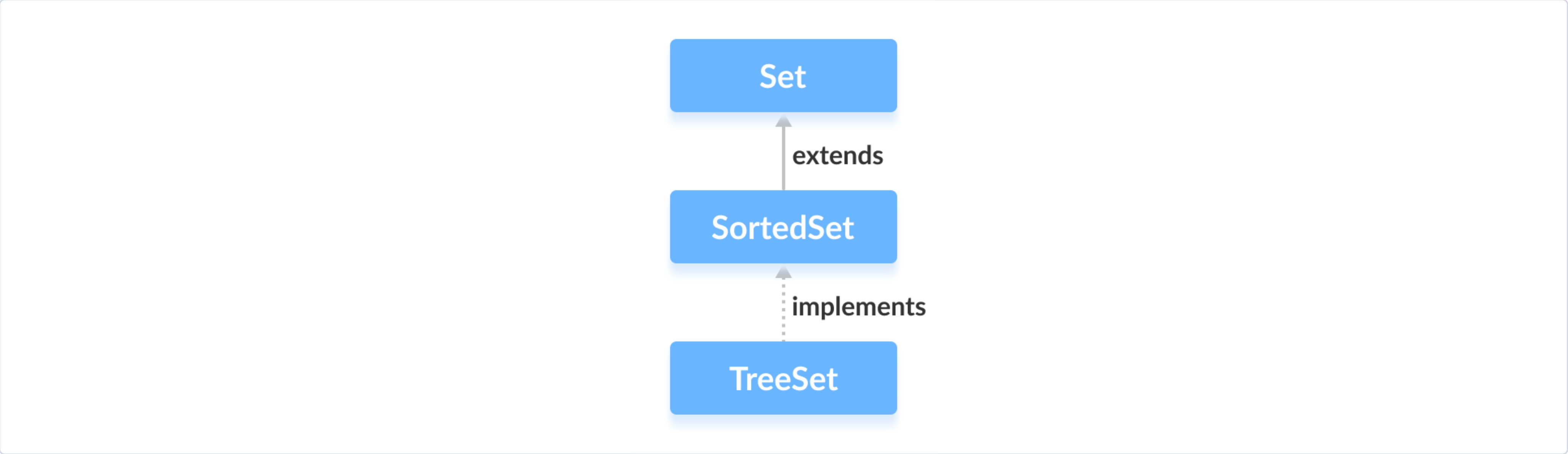
The `SortedSet` interface of the Java Collections framework is used to store elements with some order in a set.

It extends the Set interface.



Class that implements SortedSet

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



How to use SortedSet?

To use `SortedSet`, we must import the `java.util.SortedSet` package first.

```
// SortedSet implementation by TreeSet class
SortedSet<String> animals = new TreeSet<>();
```

We have created a sorted set called `animals` using the `TreeSet` class.

Here we have used no arguments to create a sorted set. Hence the set will be sorted naturally.

Methods of SortedSet

The `SortedSet` interface includes all the methods of the Set interface. It's because `Set` is a super interface of `SortedSet`.

Besides methods included in the `Set` interface, the `SortedSet` interface also includes these methods:

- **comparator()** - returns a comparator that can be used to order elements in the set
- **first()** - returns the first element of the set
- **last()** - returns the last element of the set
- **headSet(element)** - returns all the elements of the set before the specified element
- **tailSet(element)** - returns all the elements of the set after the specified element including the specified element
- **subSet(element1, element2)** - returns all the elements between the `element1` and `element2` including `element1`

Implementation of SortedSet in TreeSet Class

```java
import java.util.SortedSet;
import java.util.TreeSet;

class Main {

    public static void main(String[] args) {
        // Creating SortedSet using the TreeSet
        SortedSet<Integer> numbers = new TreeSet<>();

        // Insert elements to the set
        numbers.add(1);
        numbers.add(2);
        numbers.add(3);
        numbers.add(4);
        System.out.println("SortedSet: " + numbers);

        // Access the element
        int firstNumber = numbers.first();
        System.out.println("First Number: " + firstNumber);

        int lastNumber = numbers.last();
        System.out.println("Last Number: " + lastNumber);

        // Remove elements
        boolean result = numbers.remove(2);
        System.out.println("Is the number 2 removed? " + result);
    }
}
```

Output

```
SortedSet: [1, 2, 3, 4]
First Number: 1
Last Number: 4
Is the number 2 removed? true
```