CS112
Sets (Part 2)
Chapters 21

Lecture 18

الفصل الدراسي الثاني 1443- Spring 2022

College of Computer Science and Engineering

LinkedHashSet

- LinkedHashSet extends HashSet with a linked-list implementation that supports an ordering of the elements in the set.
- The elements in a HashSet are not ordered, but the elements in a LinkedHashSet can be retrieved in the order in which they were inserted into the set.
- A LinkedHashSet can be created by using one of its four constructors (See page 799).
- See TestLinkedHashSet.java

The SortedSet Interface and the TreeSet Class

- SortedSet is a subinterface of Set, which guarantees that the elements in the set are sorted.
- TreeSet is a concrete class that implements the SortedSet interface.
- You can use an iterator to traverse the elements in the sorted order.
- See TestTreeSet.java
- The elements can be sorted in two ways:
  - Using Comparable interface
  - Using Comparator class

Sorting TreeSet Using Comparator Class

- You need to specify a comparator for the elements in the set if the class for the elements does not implement the Comparable interface, or you don't want to use the compareTo method in the class that implements the Comparable interface.
- This approach is referred to as *order by comparator*.
- See TestTreeSetWithComparator.java page 804

Sets or Lists?

- The elements in a list can be accessed through the index. However, sets do not support indexing, because the elements in a set are unordered.
- To traverse all elements in a set, use a foreach loop.
- Compare sets and lists in terms of performance: See SetListPerformanceTest.java page 807

Tips

- If you don’t need to maintain the order in which the elements are inserted, use HashSet, which is more efficient than LinkedHashSet.
- If you don’t need to maintain a sorted set when updating a set, you should use a hash set, because it takes less time to insert and remove elements in a hash set. When you need a sorted set, you can create a tree set from the hash set.
- Java provides the static asList method for creating a list from a variable-length list of arguments. Thus, you can use the following code to create a list of strings and a list of integers:

```
List list1 = Arrays.asList("red", "green", "blue");
List list2 = Arrays.asList(10, 20, 30, 40, 50);
```