Gremaud Lucien
Paunoiu Bruno

Serie 11

IN.2028
May 14, 2019

1 Theory

a) A collection is a bunch of stuff(numbers/physical object/persons) bounded together. In Java, a collection is an object handling a bunch of same type objects through precise class methods.

b) You can add/delete one or more elements, you can determine how many objects it contains, you can check if an element is present in the collection, and you can iterate through the collection.

c) Iterator is an interface which, when implemented, helps the classes extending Collection to go through the collection object and handle each object inside of it. For example if we have a collection of strings, we could iterate through it and delete every string starting with "s".

d) Using the "for each" method we can't modify the collection(modify the elements or detele them) but this is possible if we use the iterator explicitly.

e) The elements in a Set can't appear twice and they are not in a specific order (except if it is a SortedSet) while the elements in a List can appear multiple times and they are added in a specific order.

f)

g) A Queue is a special list in which you can only add elements at the end of it and you can only remove/get the first element of the list. A Deque is similar but you can handle both ends freely.

h) The object must come from a class which implements the interface Comparable¡T¿

i) The Javadoc for the class Collection

j) Useful methods are : add(E e), contains(Object o), isEmpty(), remove(Object o), size()

2 Implementation

(d) In our implementation the ArrayList took 0.034 seconds while the LinkedList took 4.36 seconds. This is due to the fact that getting an element in an ArrayList is O(1) while in an LinkedList is O(n) because it has to jump from element to element until it gets to the one required.

3 Debugging

(a)
```
Exception in thread "main" java.util.ConcurrentModificationException
```

This exception is raised because we are removing an element of the arrayList before finishing the iteration.

(b)
```
Exception in thread "main" java.lang.IndexOutOfBoundsException:
Index: 5347, Size: 5346
```

This exception is raised because we are trying to access an index that is outside of the arrayList.

(c) The size of arrayList is modified while the loop is running, so arrayList.size() will be modified as well but since n is an integer that has been declared before the loop started, it wont be modified while the loop is running and since the arrayList n will be greater that the arrayList size.