Java Collections

Development:

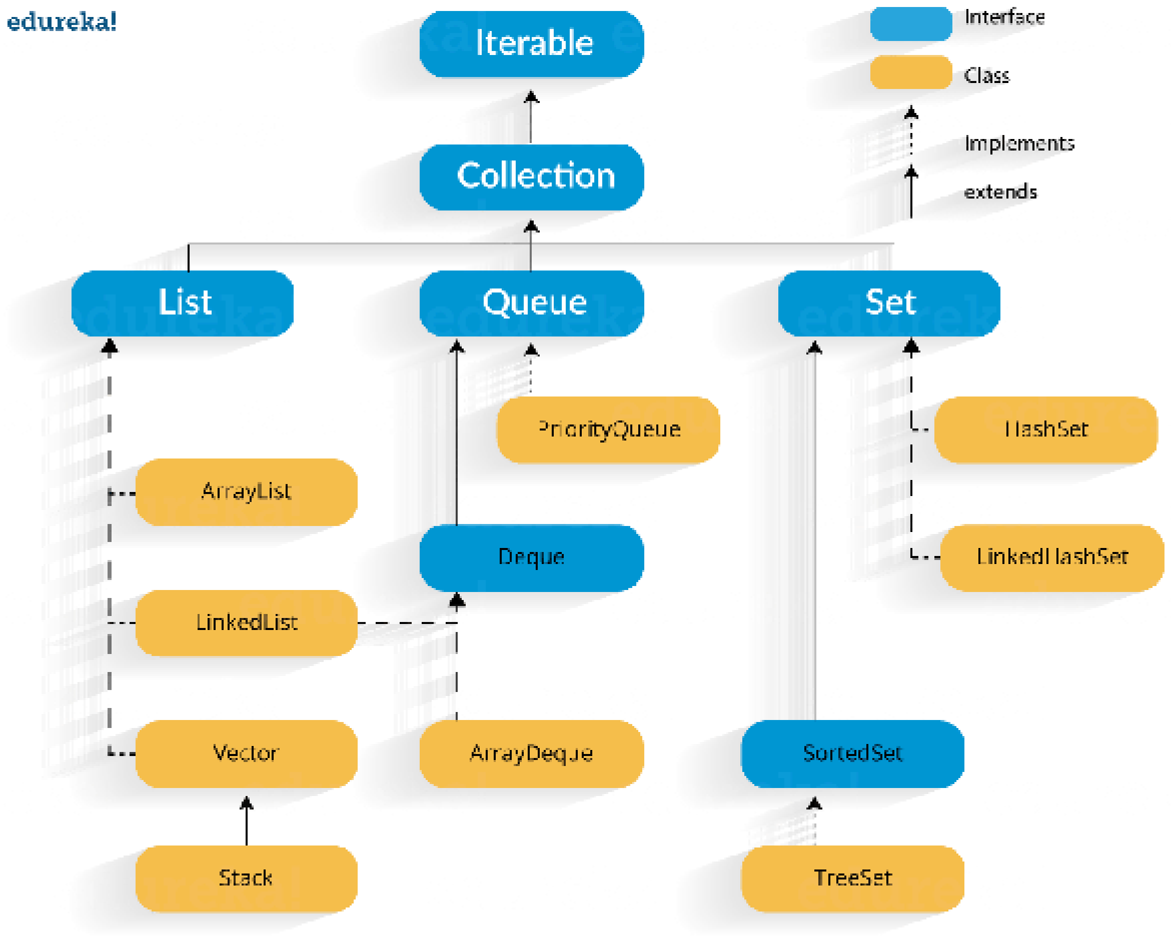
Marcelo Eduardo Guillen Castillo

Java Monterrey Academy

August 20, 2024

Java Collections

Are data structures of Java which can help us to store a variety of data and do operations on them like filters, sortering, modifications, or even store them in different manners. There exists many types of collections and in this document we will take a brief look about all of them.



Lists

A collection which stores information in a specific order can admit duplicated data.

ArrayList

```
List<Integer> arrList = new ArrayList<>();
arrList.add(1);
arrList.add(2);
arrList.add(3);
System.out.println(arrList);
```

LinkedList

A succession of data which points to the next element on the structure, it's just a big net where different nodes are connected to each other.

```
LinkedList<Integer> linkList = new LinkedList<>();
linkList.addFirst(3);
linkList.addFirst(2);
linkList.addFirst(1);
System.out.println("LinkedList");
System.out.println("> "+linkList);
```

Queue

Structures where every first element is introduced, is the first one which is removed, similar to stacks, you can only see one part of the structure, in this case only the front or “head”.

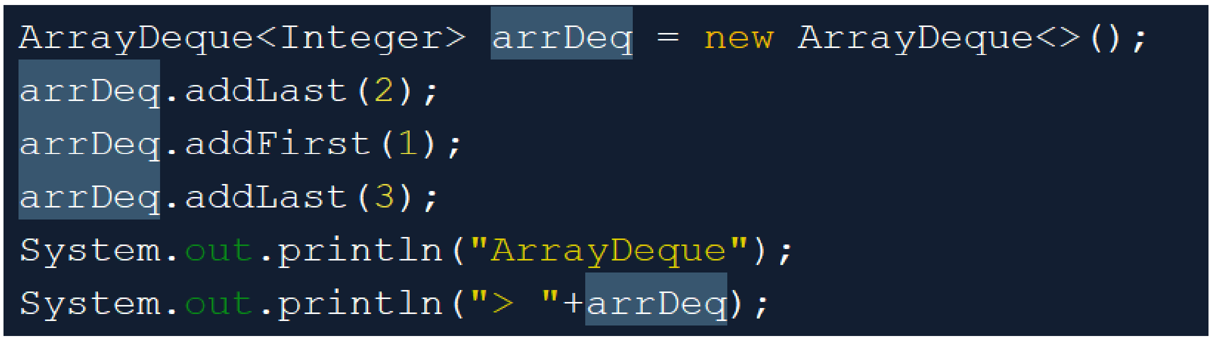
PriorityQueue

Every new data is introduced into the structure, is sorted depending on the priority of the queue, it can vary depending on the adjustments of the user.

```
PriorityQueue<Integer> prioQueue = new PriorityQueue<>();
prioQueue.add(1);
prioQueue.add(2);
prioQueue.add(3);
System.out.println("PriorityQueue");
System.out.println(prioQueue.peek());
```

ArrayDeque

A kind of special array on which the user can add or remove new data from both ends of the structure front and end, like a cards deck.



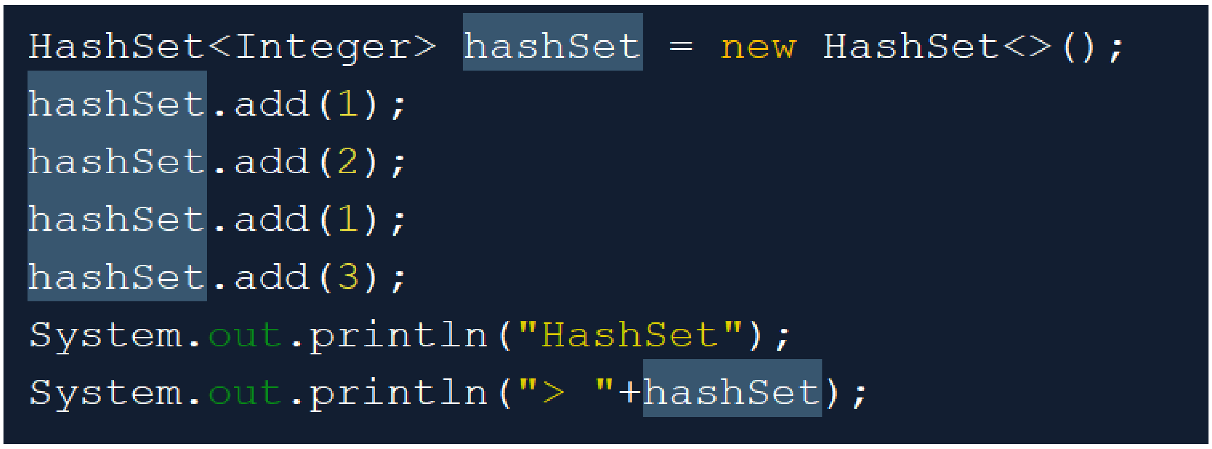
```java
ArrayDeque<Integer> arrDeq = new ArrayDeque<>();
arrDeq.addLast(2);
arrDeq.addFirst(1);
arrDeq.addLast(3);
System.out.println("ArrayDeque");
System.out.println("> "+arrDeq);
```

Set

Data structures on which its values are unique and have no order in specific.

HashSet

A collection of irrepetible and unique objects.



```java
HashSet<Integer> hashSet = new HashSet<>();
hashSet.add(1);
hashSet.add(2);
hashSet.add(1);
hashSet.add(3);
System.out.println("HashSet");
System.out.println("> "+hashSet);
```

LinkedHashSet

A sorted version of the hash sets, which maintains a double-link list through all its elements.

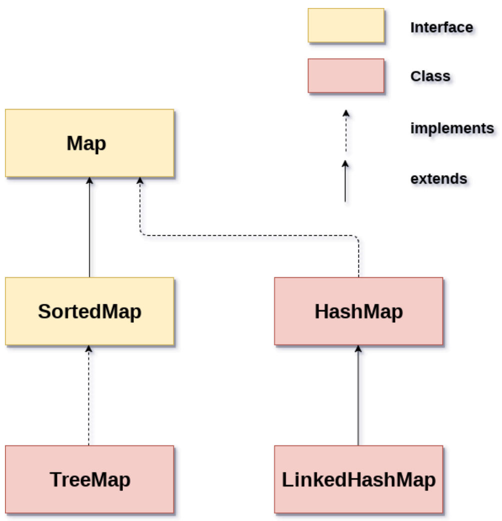
```
LinkedHashSet<Integer> lhs = new LinkedHashSet<>();
lhs.add(1);
lhs.add(2);
lhs.add(1);
lhs.add(3);
System.out.println("LinkedHashSet");
System.out.println("> "+lhs);
```

TreeSet

Stores data on a tree structure, which stores its elements in order automatically, in other cases, the order can be modified as the user wishes to be.

```
TreeSet<Integer> treeSet = new TreeSet<>();
treeSet.add(2);
treeSet.add(3);
treeSet.add(1);
System.out.println("TreeSet");
System.out.println("> "+treeSet);
```

Maps



Even if the interface isn't a Collection, it is considered as that. On this data structure we store data as value and key, the key is essential for us to access its corresponding value.

HashMap

Offers the simplest implementation of a map.

```
Map<String, Integer> mapHash = new HashMap<>();
mapHash.put("key1", 1);
mapHash.put("key2", 2);
mapHash.put("key3", 3);
System.out.println("HashMap");
System.out.println("> "+mapHash);
```

LinkedHashMap

Like a HashMap it can store elements by hashing method, but maintains an order of the elements, in this case in the insertion order.

```
Map<String, Integer> linkMap = new LinkedHashMap();
linkMap.put("key1", 1);
linkMap.put("key3", 3);
linkMap.put("key2", 2);
System.out.println("LinkedHashMap");
System.out.println("> "+linkMap);
```

TreeMap

Implementation of the Map class, where the elements are sorted according to the natural ordering of their keys.

```
Map<String, Integer> treeMap = new TreeMap<>();
treeMap.put("k1", 1);
treeMap.put("k3", 3);
treeMap.put("k2", 2);
System.out.println("TreeMap");
System.out.println("> "+treeMap);
```

References

GeeksforGeeks. (2024, 2 julio). *List Interface in Java with Examples*. GeeksforGeeks. https://www.geeksforgeeks.org/list-interface-java-examples/

W3Schools.com. (s. f.). https://www.w3schools.com/java/java_linkedlist.asp

GeeksforGeeks. (2024b, julio 24). *Stack Class in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/stack-class-in-java/

GeeksforGeeks. (2024c, julio 25). *PriorityQueue in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/priority-queue-class-in-java/

GeeksforGeeks. (2023, 14 febrero). *ArrayDeque in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/arraydeque-in-java/

W3Schools.com. (s. f.-b). https://www.w3schools.com/java/java_hashset.asp

GeeksforGeeks. (2024a, junio 19). *TreeSet in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/treeset-in-java-with-examples/

GeeksforGeeks. (2024c, julio 2). *Map Interface in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/map-interface-java-examples/