Java Collections Part 2 (Interfaces)

Java Collections Core Interfaces

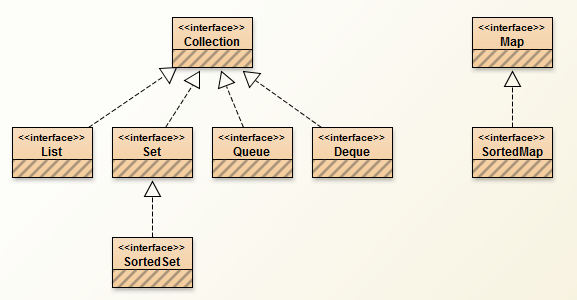


Diagram built in BlueJ.

The core collection interfaces encapsulate different collections such as List, Set, Queue and Deque. Collections operate on generics.

Collection Interface declaration

**public** **interface** Collection<E> **extends** Iterable<E>

<E> says that collection can contain any type of Object. Object can be Integer, String or any custom type Object like Customer. I will discuss generics afterwards but for now just think <E> as telling collection that E is type of Object that will be inserted in collection.

Let us see all Core collection interfaces.

**Collection** – Collection interface is root interface of all the collection interfaces. So it makes it root in hierarchy of entire Java Collections framework. It contains Objects of type E. This interface acts as base for all other interfaces in collections and also Java does not provide any kind of concrete implementation of this interface. But there exists concrete implementation of interfaces that extend Collection interface.

**Set** – Set interface represents the mathematical model of set abstraction and represents sets. It enforces the rule that any implementation of set interface MUST NOT have duplicates. Common class used is HashSet<E>.

**List** – List interface is an ordered collection. Ordered Collection means that **sequence** in which objects are inserted **is** **maintained**. Hence, we can leverage the index and get the element at index in O(1) time. Common class used is ArrayList<E>.

**Queue** – Queue interface is used to store elements prior to processing. Queue elements are not necessarily in First-In-First-Out order. The order is defined by respective implementations of concrete implementations of Queue Interface. For example LinkList<E> implements Queue<E> and it provides methods of Queue<E> interface to insert in FIFO order. Contrary to that PriorityQueue<E> inserts elements in queue based in Comparator or elements based on natural ordering.

**Deque** – Deque interface does not extend the Collection<E> interface directly. It extends Queue<E> interface. Apart from Collection<E> and Queue<E> interface functionality, it provides Last-In-First-Out. It also provided addition, deletion and retrieval of elements from both ends.

We will Discuss Map Interface in another blog.