

Interfaces of Java Collection Framework

The Collections framework has a lot of Interfaces, setting the fundamental nature of various collection classes. Let's see the most important Interfaces in the Collections framework.

- **The Collection Interface**

It is at the top of collection hierarchy and must be implemented by any class that defines a collection. Its general declaration is,

```
interface Collection <E>
```

- **The List Interface**

It extends the Collection Interface, and defines storage as sequence of elements. Following is its general declaration,

```
interface List <E>
```

1. Allows random access and insertion, based on position.
2. It allows Duplicate elements.

- **List Interface Methods**

Apart from methods of Collection Interface, it adds following methods of its own.

Methods	Description
Object get(int index)	Returns object stored at the specified index
Object set(int index, E obj)	Stores object at the specified index in the calling collection

int indexOf(Object obj)	Returns index of first occurrence of obj in the collection
int lastIndexOf(Object obj)	Returns index of last occurrence of obj in the collection
List subList(int start, int end)	Returns a list containing elements between start and end index in the collection

- **The Queue Interface**

It extends collection interface and defines behaviour of queue, that is first-in, first-out. It's general declaration is,

```
interface Queue <E>
```

Queue Interface Methods

There are couple of new and interesting methods added by this interface. Some of them are mentioned in below table.

Methods	Description
Object poll()	removes element at the head of the queue and returns null if queue is empty
Object remove()	removes element at the head of the queue and throws NoSuchElementException if queue is empty

Object peek()	returns the element at the head of the queue without removing it. Returns null if queue is empty
Object element()	same as peek(), but throws NoSuchElementException if queue is empty
boolean offer(E obj)	Adds object to queue.

- **The Set Interface**

This interface defines a Set. It extends Collection interface and doesn't allow insertion of duplicate elements. It's general declaration is,

```
interface Set <E>
```

1. It doesn't define any method of its own. It has two sub interfaces, SortedSet and NavigableSet.
2. SortedSet interface extends Set interface and arranges added elements in an ascending order.
3. NavigableSet interface extends SortedSet interface, and allows retrieval of elements based on the closest match to a given value or values.