Queue<E> Interface

Previously, I wrote on [List<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html) interface. In this post we will learn about Queue<E> interface. Queue<E> interface is a collection that is designed for holding elements prior to processing. For a Queue<E> insertion order is not strictly first in first out or FIFO. Insertion order is defined and maintained by the concrete classes that implement Queue<E>interface.

Queue<E> interface provides 3 operations i.e. Insertion, Removal and Examine. Now each of operations has 2 different methods to do same thing. Difference is one return special value(null or false) and other throws Exception.

|  |  |  |
| --- | --- | --- |
| **Operation** | **Throws Exception** | **Returns** |
| Insert | add(e) | offer(e) |
| Remove | remove(e) | poll() |
| Examine | element() | peek() |

|  |  |  |
| --- | --- | --- |
| **Operation** | **Method** | **Description** |
| Insert | **boolean** add(E e); | Inserts an element into this queue. Throws IllegalArgumentException is element cannot be added to queue. |
|  | **boolean** offer(E e); | Inserts an element into this queue. offer(e) calls add(e) in LinkedList<E>. |
| Remove | E remove(); | Retrieves and removes head of this queue. Throws Exception(NoSuchElementException) if Queue is empty. |
|  | E poll(); | Retrieves and removes head of this queue. Returns null if Queue is empty. |
| Examine | E element(); | Retrieves but does not remove head of this queue. Throws Exception(NoSuchElementException) if Queue is empty. |
|  | E peek(); | Retrieves but does not remove head of this queue. Returns null if Queue is empty. |