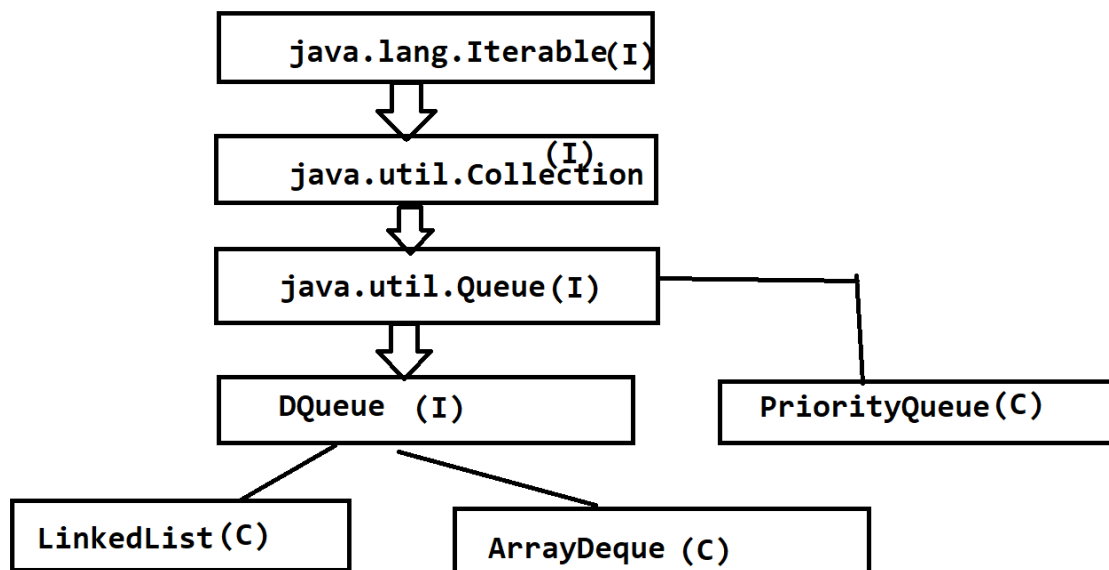## Queue Collection

The Queue interface present in the java.util package and extends the Collection interface is used to hold the elements about to be processed in FIFO (First In First Out) order. It is an ordered list of objects with its use limited to insert element at the end of the list and delete the element from start of the list i.e. It follows the FIFO or the First in First out Approach.

Following Hierarchy Represent the Queue

```
        ┌─────────────────────────┐
        │  java.lang.Iterable(I)  │
        └─────────────────────────┘
                   │
                   ▼
        ┌─────────────────────────┐
        │        (I)              │
        │  java.util.Collection   │
        └─────────────────────────┘
                   │
                   ▼
        ┌─────────────────────────┐
        │   java.util.Queue(I)    │──────────────┐
        └─────────────────────────┘              │
                   │                             ▼
                   ▼                   ┌─────────────────────┐
        ┌─────────────────────┐        │  PriorityQueue(C)   │
        │    DQueue  (I)      │        └─────────────────────┘
        └─────────────────────┘
            │           │
            ▼           ▼
  ┌──────────────┐  ┌──────────────────┐
  │ LinkedList(C)│  │ ArrayDeque (C)   │
  └──────────────┘  └──────────────────┘
```

PriorityQueue

PriorityQueue class which is implemented in the Collection Framework provides us a way to process the objects on the priority. It is known that Queue follow the First in First Out algoritham but sometimes the elements of the queue are needed to the processed according to the priority that's when the PriorityQueue comes in to play .

Example

```java
package org.techhub;
import java.util.*;
public class QueueCollectionApp {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Queue<Integer> pQueue=new PriorityQueue<Integer>();
        pQueue.add(10);
        pQueue.add(20);
        pQueue.add(15);
        //print the  top element of the priority queue
        System.out.println("Peek eleemnt  "+pQueue.peek());
        //printing the top element of the queue and removing
        System.out.println("Before Remove "+pQueue);
        System.out.println(pQueue.poll());
        System.out.println("After Remove "+pQueue);
    }
}
```