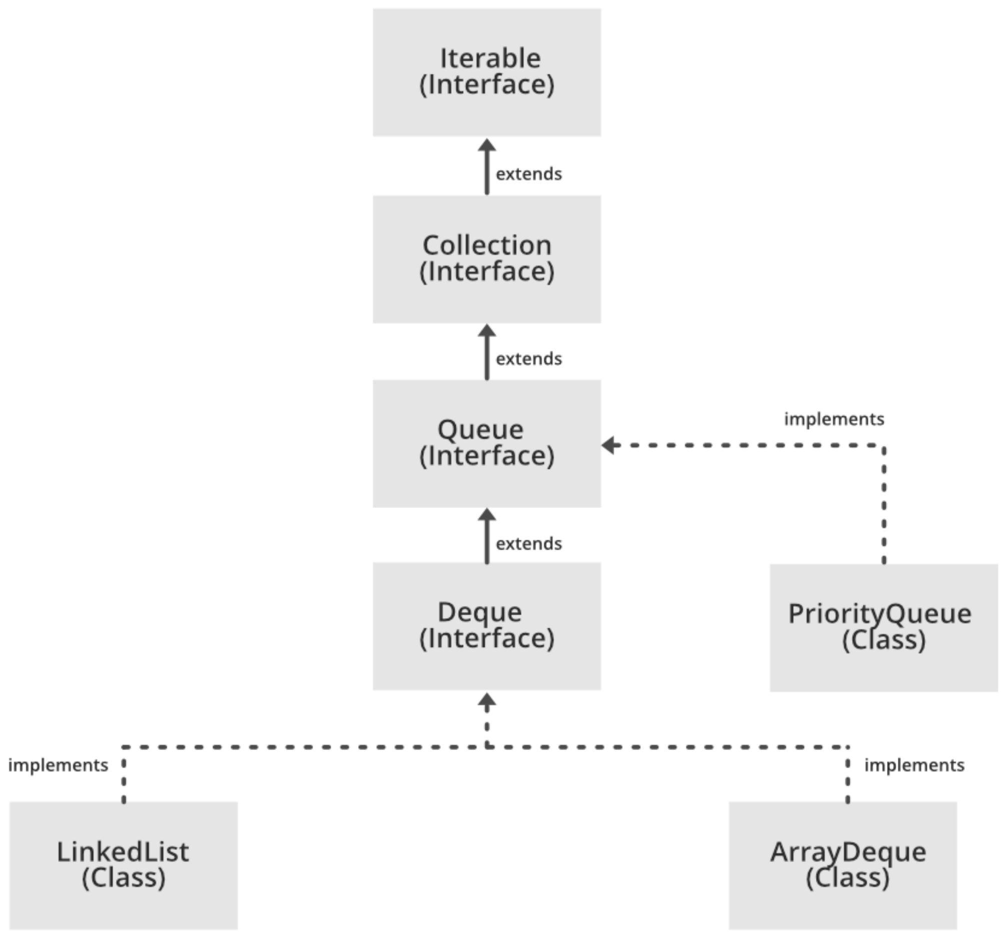
Collections in Java (Queue)

Definition of Queue:

The Queue interface is present in 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 inserting elements at the end of the list and deleting elements from the start of the list, (i.e.), it follows the **FIFO** or the First-In-First-Out principle.



Declaration: The Queue interface is declared as

```
public interface Queue extends Collection
```

Creating Queue Objects: Since *Queue* is an interface, objects cannot be created of the type queue. We always need a class which extends this list in order to create an object. It is possible to restrict the type of object that can be stored in the Queue. This type-safe queue can be defined as:

```
Queue<Obj> queue = new PriorityQueue<Obj> ();
```

Features of a Queue

- FIFO concept is used for insertion and deletion of elements from a queue.
- The Java Queue provides support for all of the methods of the Collection interface including deletion, insertion, etc.
- PriorityQueue, ArrayBlockingQueue and LinkedList are the implementations that are used most frequently.
- The NullPointerException is raised, if any null operation is done on the BlockingQueues.
- Those Queues that are present in the *util* package are known as Unbounded Queues.

PriorityQueue Class

PriorityQueue is also class that is defined in the collection framework that gives us a way for processing the objects on the basis of priority. Sometimes the elements of the queue are needed to be processed according to the priority, that's where a PriorityQueue comes into action.

public class PriorityQueue<E> **extends** AbstractQueue<E> **implements** Serializable

Example:

```
import java.util.*;
class Collection{
        public static void main(String args[]){
                PriorityQueue<String> queue = new PriorityQueue<String>();
                queue.add("Bharath");
                queue.add("Samir");
                queue.add("Devansh");
                System.out.println("Head: "+queue.element());
                System.out.println("Iterating the queue elements: ");
                Iterator itr = queue.iterator();
                while(itr.hasNext()){
```

```
                System.out.println(itr.next());
            }
            queue.remove();
            queue.poll();
            System.out.println("After removing two elements:");
            Iterator<String> itr2 = queue.iterator();
            while(itr2.hasNext()){
            System.out.println(itr2.next());
            }
        }
}
```

Output:

```
Head: Bharath
Iterating the queue elements:
Bharath
Samir
Devansh
After removing two elements:
Samir
```