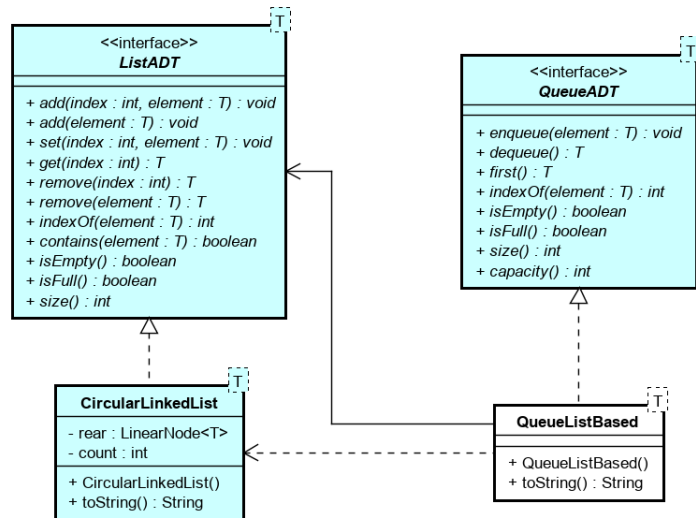## Exercise: QueueListBased – Adapter design pattern

The purpose for this exercise is to implement the adapter design pattern shown.

Note that the `enqueue` should call `add`, i.e. adding at the end (at index `size`) and `dequeue` should remove at index `0` to make the most efficient use of class `CircularLinkedList`.
Method `capacity()` may either return `Integer.MAX_VALUE` or `-1` to indicate that there is no limit.

```
<<interface>>
ListADT

+ add(index : int, element : T) : void
+ add(element : T) : void
+ set(index : int, element : T) : void
+ get(index : int) : T
+ remove(index : int) : T
+ remove(element : T) : T
+ indexOf(element : T) : int
+ contains(element : T) : boolean
+ isEmpty() : boolean
+ isFull() : boolean
+ size() : int
```

```
<<interface>>
QueueADT

+ enqueue(element : T) : void
+ dequeue() : T
+ first() : T
+ indexOf(element : T) : int
+ isEmpty() : boolean
+ isFull() : boolean
+ size() : int
+ capacity() : int
```

```
CircularLinkedList

- rear : LinearNode<T>
- count : int

+ CircularLinkedList()
+ toString() : String
```

```
QueueListBased

+ QueueListBased()
+ toString() : String
```

Class `CircularLinkedList` and interfaces `ListADT` and `QueueADT` are all given here:
- MyCollection-1.1.jar: http://ict-engineering.dk/jar/MyCollection-1.1.jar
- javadoc for MyCollection: http://ict-engineering.dk/javadoc/MyCollection/