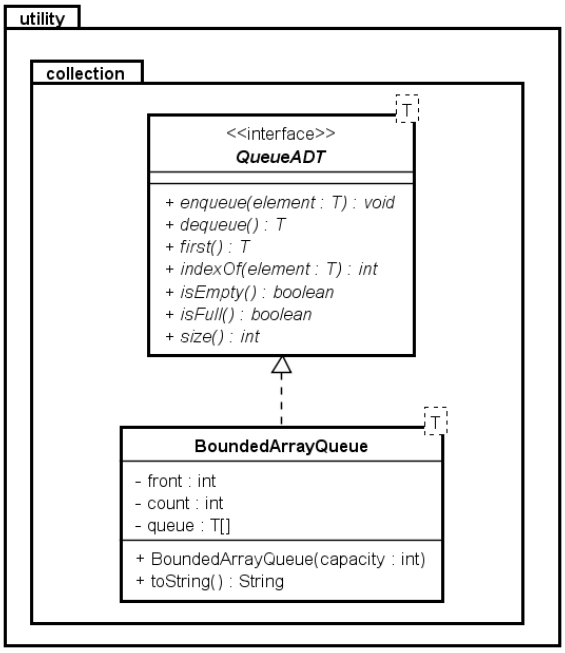
Exercises Session 2 SDJ2

Exercise 02.01 (we will reuse this later in this course)

Implement in Java exactly what you can see in the class diagram below. The diagram represents the abstract data type Queue implemented with an array as data structure. The class is called `BoundedArrayQueue`, is generic, is in package `utility.collection` and the array cannot resize. (see javadoc documentation)



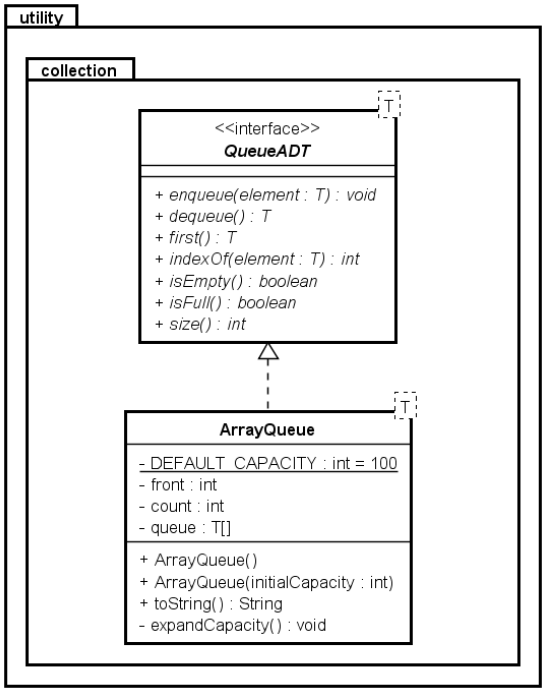
Note 1: The easy solution is to copy your solution to class `BoundedArrayStringQueue` from last session and do the following:

1. Rename the class to `BoundedArrayQueue`, make it generic with the generic type `T` and implement the interface `QueueADT<T>`, i.e.
   ```
   public class ArrayQueue<T> implements QueueADT<T>
   ```
2. Update methods by replacing e a lot of `String` return types and `String` parameters to the generic type `T`. This also goes for some temporary/local variables depending on your implementation.
3. When creating an array of the generic type do something like the following example:
   ```
   T[] myGenericArray = (T[]) new Object[75];
   ```
   In other words, replace creation of String[] with the creation of T[]

Test your solution such that you are convinced that your implementation is correct.

(Exercise 02.02)

Implement in Java exactly what you can see in the class diagram below. The diagram represents the abstract data type Queue implemented with a generic array as data structure. This time with an array that can resize when full. The implementation is somewhat similar to class `ArrayStringQueue` from last session and you just have to make it generic just like you did in the previous exercise.



Test your solution for `ArrayQueue`