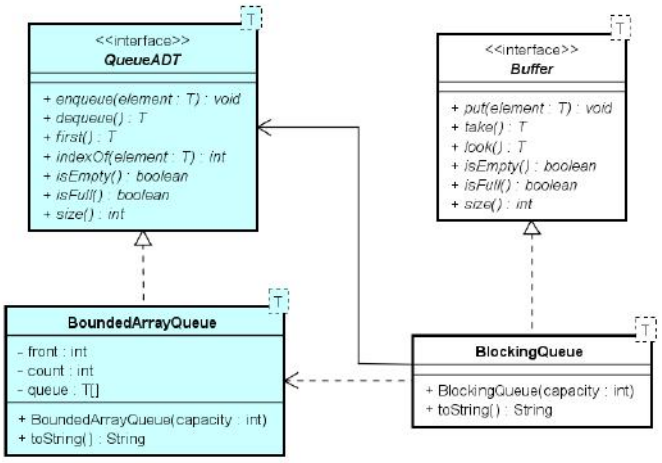
Exercise BlockingQueue – Adapter design pattern

Implement the Adapter design pattern shown – and test it.



Interface `Buffer` is like a `QueueADT` but with other names for methods (and this interface is given last in this exercise). A few remarks to the implementation (in `BlockingQueue`):

- All methods are synchronized
- Method `put` calls `wait` as long as the buffer is full (putting the calling thread into Wait State) and notifies threads waiting to take. The method throws an `IllegalArgumentException` if called with a `null` element.
- Method `take` calls `wait` as long as the buffer is empty (putting the calling thread into Wait State) and notifies threads waiting to put.
- Method `look` returns the first element or `null` if the buffer is empty.
- Methods `isEmpty`, `isFull` and `size` are exactly as in the `QueueADT`.

Class `BoundedArrayQueue` and interface `QueueADT` are given here:

- MyCollection-1.1.jar: http://ict-engineering.dk/jar/MyCollection-1.1.jar
- javadoc for MyCollection: http://ict-engineering.dk/javadoc/MyCollection/

Interface `Buffer` is given here:

```
public interface Buffer<T>
{
   public void put(T element);
   public T take();
   public T look();
   public boolean isEmpty();
   public boolean isFull();
   public int size();
}
```

a) Implement class `BlockingQueue`.

b) Test your solution, e.g. a `main` method creating a few threads putting elements into the queue and a few threads taking elements from the queue. Insert print statements in the queue, where you wait and just before you return from methods `put` and `take`. Inspect the result.