Day 17 -The Queue and Queue Interface

Chapter 04, 4.1 – The Queue, Review Problem(s): ***1, 2***

Chapter 04, 4.2 – The Queue Interface, Review Problem(s): ***4, 5***

Chapter 04, 4.3 – Array-Based Queue Implementation, Review Problem(s): ***8, 11***

Chapter 04, 4.4 – Interactive Test Driver

In Chapter 03 we presented the Stack ADT. It is based on the "Last-in, First-out" or ***LIFO*** model.

Here in Chapter 04, we will explore the Queue ADT. A Queue is based on the "First-in, First-out" or ***FIFO*** model. It's similar to standing in line at Baker…

Remember, for these ADTs, we have the high level interface, which describes the methods and what they would do, and then we consider many different implementations of the interface.

Chapter 04 PPT, pg 1-24.

True or ***False*** (provide an explanation or correction if False)

True or ***False*** The element in the queue the longest is at the rear of the queue.

True or ***False*** If you `enqueue` five elements into an empty queue and then `dequeue` five elements, the queue will be empty again.

True or ***False*** If you `enqueue` five elements into an empty queue and then perform the `isEmpty` operation five times, the queue will be empty again.

True or ***False*** The `enqueue` operation should be classified as a "setter/transformer."

True or ***False*** The element in the queue the longest is at the ~~rear~~ ***front*** of the queue.

True or ***False*** If you `enqueue` five elements into an empty queue and then `dequeue` five elements, the queue will be empty again.

True or ***False*** If you `enqueue` five elements into an empty queue and then perform the `isEmpty` operation five times, the queue will be empty again. ***`isEmpty` does not "remove" values, it just checks***

True or ***False*** The `enqueue` operation should be classified as a "setter/transformer." ***Yes, it modifies the object.***

Show what is output by the following, given that `element1`, `element2`, and `element3` are `int` variables and `q` is an object that fits the abstract description of a Queue provided in class, and you are able to store and retrieve `int` values in `q`.

```
element1=1; element2=0; element3=4;

q.enqueue(element2);
q.enqueue(element1);
q.enqueue(element1 + element3);
element2 = q.dequeue();
q.enqueue(element3 3 element3);
q.enqueue(element2);
q.enqueue(3);
element1 = q.dequeue();

System.out.println(element1+" "+element2+" "+element3);

while ( !q.isEmpty() )
{
   element1 = q.dequeue();
   System.out.println(element1);
}
```

```
element1=1; element2=0; element3=4;

q.enqueue(element2);                 0
q.enqueue(element1);                 0, 1
q.enqueue(element1 + element3);      0, 1, 5
element2 = q.dequeue();              1, 5
q.enqueue(element3 * element3);      1, 5, 16
q.enqueue(element2);                 1, 5, 16, 0
q.enqueue(3);                        1, 5, 16, 0, 3
element1 = q.dequeue();              5, 16, 0, 3

System.out.println(element1+" "+element2+" "+element3);
1 0 4

while ( !q.isEmpty() )
{
   element1 = q.dequeue();
   System.out.println(element1);
}
5, 16, 0, 3
```

Lab-time, time permitting. ***Expected to stay unless ALL outstanding work is completed.***