blocking queues & spoolers

This work investigates the use of blocking queues.

Follow through the steps below.

stages

1. write a class which either wraps or extends a blocking queue. The queue will be used to handle textual messages;
2. write a DataProcessor interface with contract for...
   void processData(String)
3. extend the wrapper (from stage 1) in the following ways...
   a. to include an addDataProcessor method (it only needs to handle at most one dataProcessor);
   b. so the constructor builds a thread which dequeues an item & passes it to the DataProcessor (if there is one);
4. write a DataProcessor which handles messages containing the type of information supplied to the clone/non-clone test agents used in the lecture series (word, delay & count) & processes them in a similar loop;
5. write a suitable test harness so you can post messages on the queue (you could use the NetBeans GUI designer);
6. extend the test harness so it deals with two queues (q1 & q2) each with their own DataProcessor. The test harness should allow the user to post messages on q1; q1 does some message handling directly and also passes some (partial?) requests onto q2 (invent any behaviour for q1 & q2 which works for the test).

advanced work

1. you have probably developed stages 1-6 to perform in a similar way to the non-clone agent. Experiment with a second type of queue wrapper which behaves as a cloning agent;
2. investigate with your queue wrapper so it can be associated with multiple DataProcessors – what are the advantages & disadvantages in this kind of facility (both generally and also for queue DataProcessors in particular).