Part 2 Java:

My Implementation uses a main method to instantiate a monitor a producer and N number of consumers. The consumers and producers make calls to the monitor whenever they want to modify the queue. Three parameters are taken on input to decide number of consumers(n), max request length(l), and producer sleep time(s). More parameters could be taken in for how many jobs a producer should make or how many jobs a consumer should take. For now each consumer will terminate after taking 3 jobs, and the producer will terminate after producing 3*N jobs.

wait() and notify() are used to communicated between the monitor and the threads. Producers will wait if the queue is full, and notify if the queue is no longer empty. Consumers will wait if the queue is empty and will notify if the queue is no longer full.

The requests are int[] that hold a randomized ID and randomized length between 1 and max length. These are held in a Queue of int[]. When a request is made, the producer must sleep for "s" seconds before it will try to produce another. Monitor dequeue needed to return the job so that the consumer thread knew how long it needed to wait.

-Matthaus Wolff