COL106 Assignment2
Name: Dipen Kumar
Entry No: 2018CS50098
Group Number: 3

Queue.java

Queue class implements QueueInterface.
Its constructor takes an int type argument which denotes the size of the queue. I have initialized the currentSize=0 and front=0 and rear=0.
Each time when I enqueue element in the queue, I insert it at rear and update rear=rear+1. Similarly each time when I dequeue element in the queue, I return the element at front index and update front=front++;
Also for every enqueue I increase the currentSize by 1 and for dequeue I decrease it by 1.
Note: With the help of modulus(%) I ensure that the value of rear and front don't exceed capacity-1. Each time I increase front and rear I store its value obtained after taking its mod.
e.g. rear=(rear+1)%capacity;
We are ensuring this because we have used a array of length capacity. We can store from index i=0 to i=capacity-1.
After arriving at index capacity-1, the next index it will go is 0 and 1 and 2 and so on.
Also if the queue is full i.e. queue.size()==capacity. In this case it will not enqueue any element
Similarly if queue is empty i.e queue.size()==0, it will return null when dequeue is called.
Here size() method returns currentSize.

PriorityQueue.java

Same thing what I did in Queue is done here with some extra feature. That extra feature is sorting which I am doing at the time of enqueue. After I enque a new element in a priority queue in the same way I have done in queue, I sort all its elements in their priority order. Element with low priority value comes first i.e. towards front index while the one with high priority value come latter i.e. towards the rear. In this way I ensure that whenever I dequeue an element it will return element at index=front but since it it sorted at the time of enqueue, element at index=front is element with low priority value. That is what we want from priority queue. I have used bubble sort to achieve this sorting.

Node.java
This class has two public method getPriority() and getValue() which return value in field “priority” and “value” of data type <int> and generic type <V>.

Buyer.java
This class extends BuyerBase class. Its constructer initializes “catalog” filed with the priority queue from where buyer buy items and where to seller put items from inventory (queue). It also initialize its lock of type Lock with the common lock created in Assignment2Driver.java. Same lock was passed to seller.java. Also it initializes two contidion full and empty created in Assignment2Driver.java. SleepTime and Iteration were initialized by calling setSleepTime () and setIteration() method, since these were two private field decleared in BuyerBase class. There is a buy() method in this class which dequeue item from catalog once acquire the lock. It also check for whether the catalog is empty or not. If it is empty it wait for signal empty.signalAll() and then dequeue catalog and signal full.signalAll which implies that now there is space in catalog. Seller can add their items to catalog if they were waiting. Finally it release the Lock lock.

Seller.java
Similar to Buyer.java, Seller.java extends SellerBase<V> class. Its constructer initializes catalog (priority queue), lock (Lock), full (Condition), empty (condition), SleepTime (int), inventory (queue) and catalogSize (int).
Lock and Condition was created in Assignment2Driver.java and was passed here, same was passed to Buyer.java.
It has sell() method which acquire the lock and dequeue one item from inventory and enqueue this in catalog. It also check whether the catalog is full or not. If it is full then it wait till signal full.signalAll() and then dequeue inventory and enqueue catalog and signal empty.signalAll() so that if Buyer was waiting to buy since no items were in catalog earlier can now buy item.
Also this will continue till the time there is an element in inventory.
Finally it releases the Lock lock.

Assignment2Driver.java
In this starter code we were supposed to do two things- (rest were done)
First, we were supposed to add all elements of the ArrayList named "list" to inventory queue. List contained nodes with a item and its priority. I achieved this by iterating over list for(int i=0;i<list.size();i++) and during each iteration I was enqueuing the node that was at ith index in list to inventory.

Second we were supposed to create multiple Buyer and Seller Threads and start them.
Again I used for loop from 1 to numBuyers and each time loop executed I make a new buyer Thread and start it.
Same for Seller Thread, I used for loop from 1 to numSellers and each time loop executed I make a new Seller Thread and start it.

Bases.java, Item.java and PriorityQueueTestDriver.java
These .java files were given to us from instructor side and we were asked not to change anything in these files. These files were not edited by me.

Thank You…