

Explain your implementation:

The two important threads are the Main thread and the Door thread. The door thread periodically spawns (and starts) customer threads while the bar is open (3 seconds). Customers will try to enter the Serving Area (which is on the main thread). If they fail, they will wait until notified. They will then proceed to order their meal (which lets SA track statistics), sleep for a while to simulate eating time, and finally leave.

The serving area has synchronized methods for the customer to call. This ensures that the shared variables have their proper state at all times.

What is the functionality of wait(), notify(), and notifyAll()?

wait()

Causes the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.

notify()

Wakes up a single thread that is waiting on this monitor.

notifyAll()

Wakes up **all** threads that are waiting on this monitor

What is the difference between notify() and notifyAll()?

notify will only wake up a single thread, while notifyAll will wake up **all** waiting threads.

In some cases, all waiting threads can take useful action once the wait finishes. An example would be a set of threads waiting for a certain task to finish; once the task has finished, all waiting threads can continue with their business. In such a case you would use notifyAll() to wake up all waiting threads at the same time. When this is not the case, it would be wasteful to wake up a ton of threads when only one of them will do any work.

Which variables are shared and what is your solution to manage them?

The main shared resource is the ServingArea's list of eating customers. The stats counters (takeawayOrders and eatHereOrders) are also shared variables.

We are protecting these shared resources by making them private and using synchronized access methods.

Which method or thread will report the statistics and how will it recognize the proper time for writing statistics?

Statistics should never be printed before the door closes. We have a blocking method on ServingArea that prints the statistics given that the door is closed and no customers remain. We call this from the Door thread after we're done spawning customers.