

Synchronization Challenge, The Shoe Fulfillment Warehouse

In this challenge, you'll be creating your own Producer Consumer example, for a Shoe Warehouse Fulfillment Center.

The producer code should generate orders, and send them to the Shoe Warehouse to be processed.

The consumer code should fulfill, or process the orders in a FIFO or first in, first out order.

You'll be creating at a minimum, three types for this, an Order, a Shoe Warehouse, and a Main executable class.

The Order

An Order should include an order id, a shoe type, and the quantity ordered.

A record might be a good fit for this type.

The Shoe Warehouse

The shoe warehouse class:

- Should maintain a product list, as a public static field.
- It should also maintain a private list of orders.
- It should have two methods, `receiveOrder` and `fulfillOrder`.
- The `receiveOrder` gets called by a Producer thread. It should poll or loop indefinitely, checking the size of the list, but it should call `wait` if the list has reached some maximum capacity.
- The `fulfillOrder` gets called by a Consumer thread. It should also poll the list, but it needs to check if the list is empty, and wait in the loop, until an order is added.
- Both methods should invoke the `wait` and `notifyAll` methods appropriately.

The Applications main class and method

Finally, you'll need some kind of a Main class with a main method, to execute.

- This method should create and start a single Producer thread. This should generate 10 sales orders, and call `receiveOrder` on the Shoe Warehouse, for each.
- In addition, you'll create and start two Consumer threads. Each thread needs to process 5 fulfillment orders, calling `fulfillOrder` on the Shoe Warehouse for each item.

You'll test your Producer Consumer application, and confirm your application fulfills all the 10 orders it receives.