

## Writing Threads in java

This project is meant to give you a taste of coding the creation and execution of threads, and with the use of the Thread class methods. In order to synchronize the threads you will have to use (when necessary), `run( )`, `start( )`, `currentThread( )`, `getName( )`, `join( )`, `yield( )`, `sleep(time)`, `isAlive( )`, `getPriority( )`, `setPriority( )`, `interrupt( )`, `isInterrupted( )`, synchronized methods

The fishermen and the elusive “big catch”

Lake Big Fish is known for its very large fish. There is a town on the shore called Breton. Near Breton is an island called Marrowind with a fishing hole where the large fish often come to feed. This fishing hole has become famous among amateur fishermen who come from all around to fish in the famous fishing hole. There is a shop in Breton that will stuff the prize and mount it for the fishermen so they can come home with proof of their “Big Catches”. The fishermen can also bring in lesser fish for money. And use very small fish for bait. Every fisherman (**num\_fm**) will leave once he has at least one “Big Catch” and enough money to pay for his trip, **vacation\_cost**.

When a fisherman has a fish big enough to mount or cash in for money he takes a boat to the town and goes to the shop (use `sleep()` to simulate the boat trip). When he gets to the shop he sets his variable **needs\_help(i)** to true and **busy waits** until a clerk is ready to help him. There are **num\_ca**, customer associates but only one line of fishermen. A customer associate should pick the next waiting fisherman in a mutual exclusion fashion (this can be done from inside of a **synchronized** method) and in a FCFS order.

A fisherman that has not achieved his “Big Catch” or not recouped his **vacation\_cost** will travel to the island to get his turn at the fishing hole. Fishermen gather on the island of Morrowind and **wait** (simulated by **sleeping** a long time) for the fishing hole to be available. The fishing hole is guarded by a Ranger. The ranger randomly interrupts one of the fishermen (use `interrupt()` and `isInterrupted()`). The ranger allows the fishermen access to the fishing hole and allowed one turn to fish. If the fisherman gets no fish, the Ranger allows the fisherman to increase his priority for a very short time and fish one more time. (use `getPriority( )`, `setPriority( )` and `currentThread( )` methods). After that time at the fishing hole the fisherman will immediately reset his priority back to the default value and will allow another fisherman to use the fishing hole (use `yield( )`). If the fisherman is successful he will receive a random fish (bait, 10, 20,30,40,50 or 200 pound fish) If the fisherman does not have at least 10 pounds of fish he must wait for the ranger to give him access to the fishing hole again. When he gets the 10 pounds of fish he can go back to the shop in Breton and get his mounted fish and or his money for the other fish (\$0.75 per pound of fish)

One the fisherman achieves his goal of a big fish and **vacation\_cost**. He will go home. However, each fisherman must join the previous fisherman in sequence. For example, in case that fisherman (i+1) is alive, fisherman (i) will join fisherman (i+1) (use `isAlive( )` and `join( )`). The last fisherman to leave for home (fisherman(0) ) will let the Ranger know that it is time for him to terminate. The customer associates will terminate as well.

Using Java programming, synchronize the 3 types of threads, Fisherman, Customer associate and Ranger, in the context of the problem. Follow the requirements as specified.

The number of fishermen,(num\_fm) , number of customer associates, and the vacation\_cost must be entered as command line arguments. The default value for the number of fishermen is 6 and the vacation cost is \$250 and number of customer associates is 2.