

Threads

Purpose

To become more familiar with using threads.

You will need to download the file ThreadPractice.zip from Pilot for this lab. This code involves one taskmaster thread and several worker threads. The taskmaster thread creates new tasks and adds them to the order queue. Each task takes a certain number of milliseconds to complete. The worker threads pick up tasks from the order queue and simulate working on them by sleeping for the specified number of milliseconds. The program continues until a total of 100 tasks have been created.

Part 1: Right now the queue can behave inconsistently. For example, occasionally two workers can both accept the same task (see below, but keep in mind that threads are non-deterministic, so your output will likely be different each time you run the program). Modify the program so that this cannot occur.

```
Worker 3 accepted Task 1: 1007 ms
Worker 1 accepted Task 2: 896 ms
Worker 2 accepted Task 1: 1007 ms
```

~~**Part 2:** Right now the program can say it is finished before all of the tasks have been made and completed (see below). Modify the program so that this cannot occur.~~

```
Worker 1 accepted Task 1: 1175 ms
Finished?
Worker 2 accepted Task 2: 765 ms
```

Part 3: Right now if a worker requests a task to work on but the queue is empty, the acceptTask method will loop until the queue is no longer empty, as shown below. This is called a “busy loop” and it is not an efficient use of CPU time. Modify this code so that the method instead waits until it has been notified that a new task has been created.

```
public Task acceptTask() {
    while (orders.isEmpty()) {
        // looping until there is a task in the queue to accept
        System.out.println("waiting...");
    }
    return orders.poll();
}
```

Part 4: Change the program so that if there are five tasks in the queue, the program will wait until the number of tasks falls below five before adding any more tasks to the queue.

Hint: You can solve all of the problems above by only making changes to two files: Driver.java and OrderQueue.java