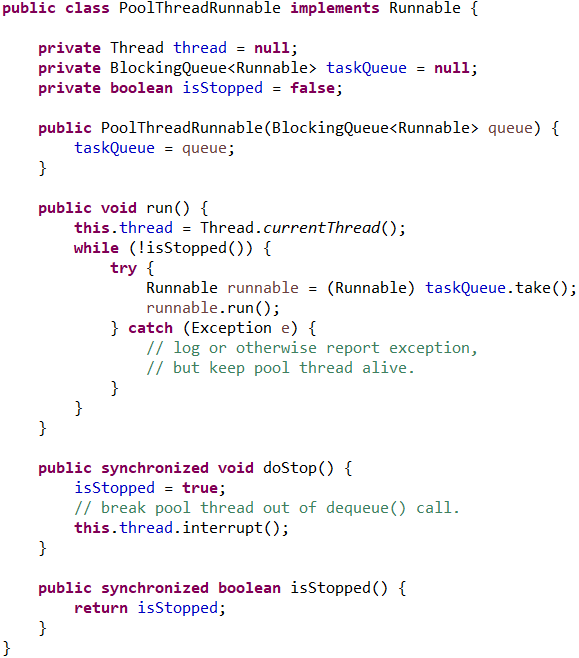
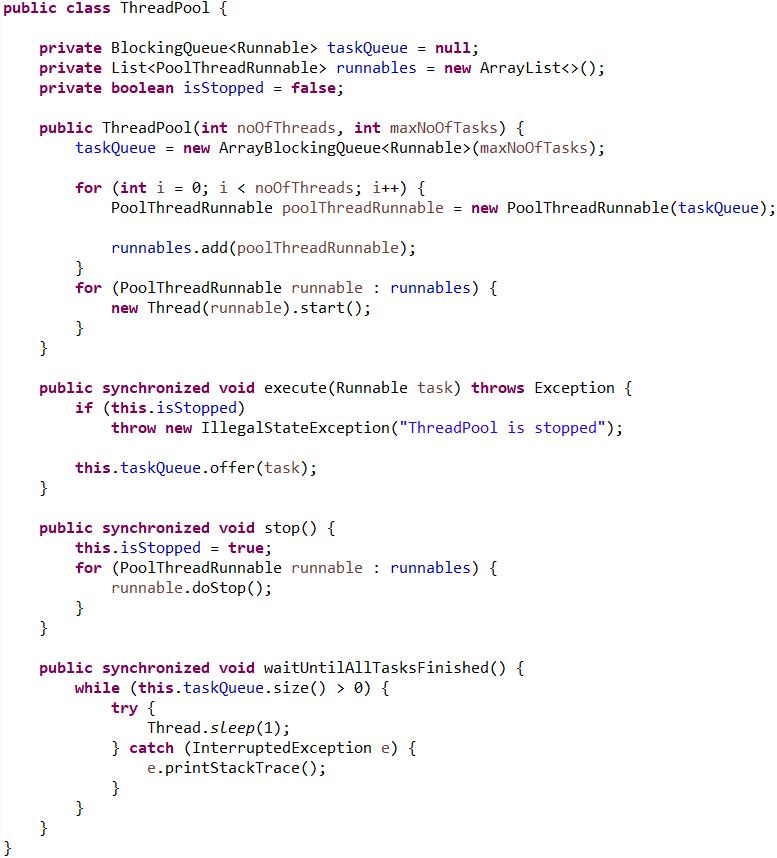
**Laboratory 9**

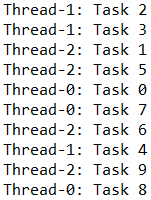
1. I implemented the Thread Pool example as follows:







The result for running this code is:



The previous code has 10 tasks and 3 threads, all of them being stored in the ThreadPool object.

In the ThreadPool is initialized the blocking queue with its size in which the tasks to execute will be stored, and also the thread list with all the threads that can be used at execution in the ThreadPool.

When calling the method execute(Runnable task) from a ThreadPool object, the task given as a parameter will be added in the queue of tasks that wait to be executed by the threads in the ThreadPool. The method waitUntilAllTasksFinished() waits until the queue of tasks to be executed becomes empty. The method stop() will stop the execution of ThreadPool instance and all the threads in it.

Each thread runs tasks till there are no tasks in the task queue or it is stopped by the ThreadPool by calling the method doStop().

In the main is created the ThreadPool instance and are also created the tasks as runnables and added in the queue by calling the method execute() of ThreadPool object for each runnable. After all tasks are finished, the program will stop.