import java.util.concurrent.TimeUnit;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

public class ReentrantLockImplementation implements Runnable {

final Lock lock = new ReentrantLock();

String name;

Thread t;

public void createThreads(String threadname) {

name = threadname;

t = new Thread(this, name);

System.out.println("Creating New Thread: " + t.getName());

t.start();

}

public static void main(String args[]) {

ReentrantLockImplementation obj = new ReentrantLockImplementation();

obj.createThreads("Thread 1");

obj.createThreads("Thread 2");

obj.createThreads("Thread 3");

}

public void run() {

do {

try {

if (lock.tryLock(400, TimeUnit.MILLISECONDS)) {

try {

System.out.println(Thread.currentThread().getName()+" acquired the lock. "

);

Thread.sleep(1000);

} finally {

lock.unlock();

System.out.println(Thread.currentThread().getName()+" released the lock. "

);

}

break;

}

else

{

System.out.println( Thread.currentThread().getName() + " could not acquire the lock. Need to try for the lock again.");

}

}

catch (InterruptedException e)

{

System.out.println(e);

}

} while (true);

}

}