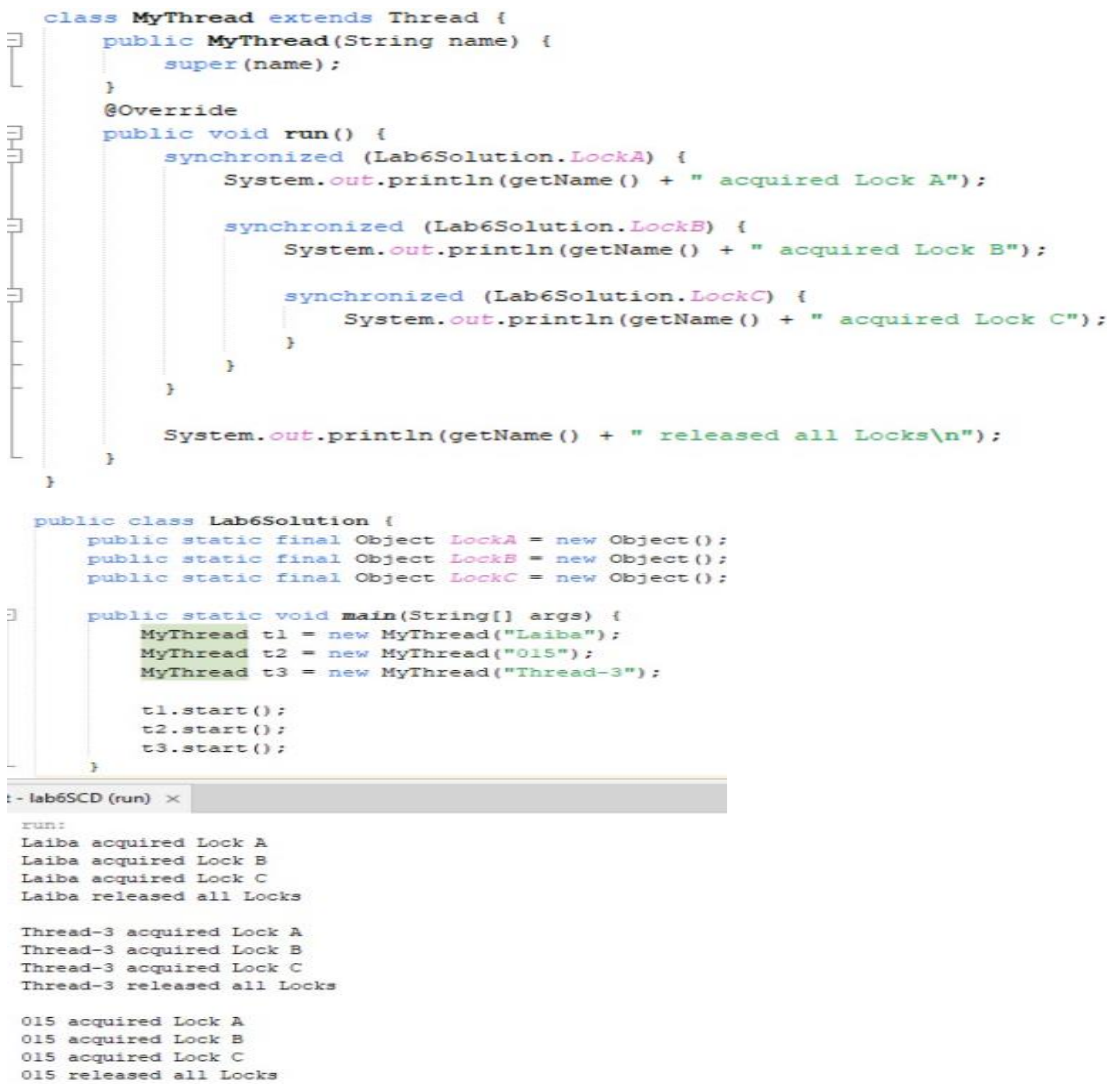
LAB # 6
Deadlock in concurrency

OBJECTIVE:

Implementing multiple thread blocked resources with help of lock and deadlock conditions.

Lab Task:

Create three threads by implementing thread synchronization block through 3 locks. (Hint: Apply un-sequenced lock to analyze deadlock and solve it through provided solution:



```
class MyThread extends Thread {
    public MyThread(String name) {
        super(name);
    }
    @Override
    public void run() {
        synchronized (Lab6Solution.LockA) {
            System.out.println(getName() + " acquired Lock A");

            synchronized (Lab6Solution.LockB) {
                System.out.println(getName() + " acquired Lock B");

                synchronized (Lab6Solution.LockC) {
                    System.out.println(getName() + " acquired Lock C");
                }
            }
        }

        System.out.println(getName() + " released all Locks\n");
    }
}

public class Lab6Solution {
    public static final Object LockA = new Object();
    public static final Object LockB = new Object();
    public static final Object LockC = new Object();

    public static void main(String[] args) {
        MyThread t1 = new MyThread("Laiba");
        MyThread t2 = new MyThread("015");
        MyThread t3 = new MyThread("Thread-3");

        t1.start();
        t2.start();
        t3.start();
    }
```

: - lab6SCD (run)

```
run:
Laiba acquired Lock A
Laiba acquired Lock B
Laiba acquired Lock C
Laiba released all Locks

Thread-3 acquired Lock A
Thread-3 acquired Lock B
Thread-3 acquired Lock C
Thread-3 released all Locks

015 acquired Lock A
015 acquired Lock B
015 acquired Lock C
015 released all Locks
```