Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Synchronization in Java (Part 3)

=> Disadvantages of Synchronization :-

1. It will decrease the application performance because the threads has to wait for the object lock
2. Due to synchronization "deadlock" condition may occur

=> Deadlock :-

➔ Deadlock is the situation where one thread is waiting for an object lock that is acquired by another thread and second thread is waiting for an object lock which is acquired by first thread. In this type of situation both threads will wait for each other locks to release and this situation goes for infinite time, thus deadlock condition will occur

---

=> Problems with "synchronized" keyword :-

1. If we want to acquire the lock in one method and release the lock in another method then this cannot be dont with synchronized keyword
2. There is no flexibility to try for the lock without waiting
3. There is no way to get the list of all the waiting threads
4. We cannot judge/control which thread will acquire the lock when the first thread will release the lock

=> To overcome these problems java introduced java.util.concurrent.locks package (Lock interface & ReentrantLock class) in JDK 1.5 version

=> Lock interface :-

- ➔ Lock interface is used to achieve synchronization same as synchronized blocks
- ➔ Syntax :

```
        public interface Lock
        {
                void lock();
                void unlock();
                boolean tryLock();
                boolean tryLock(-) throws
InterruptedException;
        }
```

=> ReentrantLock class :-

- ➔ It is the implemented class of Lock interface.
- ➔ Syntax :-

public class ReentrantLock implements Lock, Serializable

{

//constructors :-

1. public ReentrantLock();

2. public ReentrantLock(boolean fair);

// contains all the methods of Lock interface

final int getHoldCount();

final boolean isLocked();

public boolean isHeldByCurrentThread();

```
        public final int getQueueLength();
    }
```

=> Note :-

1. We should always use unlock() method in finally block
2. If we use lock() method multiple times then we have to unlock it multiple times only

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming