Jenkov

<http://tutorials.jenkov.com/java-concurrency/nested-monitor-lockout.html>

**一、Deadlock**

<https://github.com/BlankKelly/ConcurrencyNote>

1 什么是死锁

多个线程竞争**多个资源**，导致**一直阻塞**

误区：

1 synchorized 只允许一个线程去访问，不会产生死锁。

如果对象在调用同步方法的时候，又调用了这个类中其他对象，则需要获得两个锁才能执行，可能产生死锁。

## More Complicated Deadlocks

## Database Deadlocks

数据库中因为多个事务产生死锁。

**Deadlock Prevention**

## Lock Ordering

多所有的锁进行排序，申请的时候按照次序来申请。

## Lock Timeout

1 对每个锁的申请都有一个过期事件， 如果超过过期时间，则放弃已经拥有的锁，并休眠一段随机的时间，然后重新获取。

## Deadlock Detection

死锁监测很重，在不能使用Lock Ordering情况下，lock timeout 不可行时。

# Starvation and Fairness

## Causes of Starvation in Java

1 因为线程优先级不同导致的线程饥饿

不好不要设置线程优先级。

2 因为获取不到锁导致的线程饥饿

当多个资源竞争一个锁时，恰好有一个线程总是竞争不到.

->使用公平锁。（可能导致性能下降： 线程执行的频繁程序（越频繁影响越大、线程执行任务时长（时间越短，影响越大））

3 因为线程通知不到notify（随机通知一个线程）导致的线程饥饿。

# Nested Monitor Lockout

内置监视器失效：

内置监视器在外部之内；只是放弃了内部的锁；但是要改变状态，其他线程需要获得外部的锁。

# Slipped Conditions

# Locks in Java

# 二、Java IO Tutorial

# 三、Maven

**Build Life Cycles, Phases and Goals**

**生命周期由一系列的phases组成，每一个phase由于一系列goals组成。**

**Dependencies and Repositories**

如果本地仓库（指定的文件路径）没有依赖包，会从中心仓库中下载到本地仓库。

**Build Plugins**

standard Maven build phases and goals 没有提供的额外动作，可以在pom文件中配置插件

**Build Profiles**

为了不同的目的：1 本地开发测试 2 远程部署

## Maven POM Files

1 首先是GAV

groupId: 仓库的路径

artifactId： 仓库路径+ 打包后的文件名

version： 仓库路径+打包后的文件名

### Super POM

可以设置parent来继承父pom，子pom可以重写父pom的设置。

### Effective POM

## Maven Settings File

两个配置文件：settings.xml

## Running Maven

mvn command you pass the name of a [**build life cycle, phase or goal**](http://tutorials.jenkov.com/maven/maven-tutorial.html#maven-build-life-cycles-phases-and-goals) to it

1 mvn install : install是phase

2 mvn clean install : clean是生命周期

3 mvn dependency:copy-dependencies dependency是phase，copy-dependencies 是goal

**Maven Directory Structure**

## Project Dependencies

### External Dependencies

不在仓库路径下的依赖包，可以使用systemPath设置路径

<systemPath>${basedir}\war\WEB-INF\lib\mydependency.jar</systemPath>

**Snapshot Dependencies**

在版本号后面追加 -SNAPSHOT，即使本地仓库有，也会从远程仓库中下载。

## Maven Repositories

* Local repository
* Central repository
* Remote repository

## Maven Build Life Cycles, Phases and Goals

**Build Life Cycles**

1. default：项目部署、phase 顺序执行
2. clean： 项目清理
3. site：文档生成

## Maven Build Profiles

在不修改pom文件的情况下，替换里面的配置。

使用profile中定义的配置替换pom文件中的配置来进行执行。

## Maven Plugins

# Java JSON Tutorial