课程地址：https://coding.imooc.com/learn/list/409.html

## 开宗明义【告诉你：学什么+收货什么】

## 鸟瞰Java并发【上帝视角建立并发知识框架】

### 并发工具纵览-建议其 Java 并发体系的大厦

这个概览的章节，学完以后可以再来看

》》并发工具类-三大类，有如下三大类

A screenshot of a cell phone

Description automatically generated

》》具体每一个类的地址

A screenshot of a cell phone

Description automatically generated

前两面两个模块都是为了安全，从不同的角度来看的。

由点成线，由线成面，需要的就是要理解深刻。

》》第二大部分：管理相关的

A picture containing map

Description automatically generated

》》第三部分：线程之间配合的

A picture containing screenshot, text

Description automatically generated

AQS

### 课程各章节思维导图汇总

Java并发成神之路——JUC全方位详解：

https://naotu.baidu.com/file/89fb28b05e3395800f9dc2d332d2b198?token=9b45e08e55281667

《Java并发核心知识体系精讲》：完整清晰的并发知识网络+Java内存模型+高频面试题详解

https://coding.imooc.com/class/362.html

线程8大核心基础：

http://naotu.baidu.com/file/07f437ff6bc3fa7939e171b00f133e17?token=6744a1c6ca6860a0

Java内存模型——底层原理：

http://naotu.baidu.com/file/60a0bdcaca7c6b92fcc5f796fe6f6bc9?token=bcdbae34bb3b0533

死锁——从产生到消除：

http://naotu.baidu.com/file/ec7748c253f4fc9d88ac1cc1e47814f3?token=bb71b5895a747d67

## 线程池【治理线程的最大法宝】

### 线程池的自我介绍

### 增减线程的时机

### keepAliveTime 和线程工厂

### 演示内存溢出的情况

### 线程池用法演示

### 对比线程池的特点

### 五虎上将收服线程池

### 钩子：给线程池加点料

### Executor 家族的辨析

### 线程池状态和使用注意点

## ThreadLocal【一次解决老大难问题】

### ThreadLocal 的两种用法

### 进化之路：故事从两个线程说起

### 进化之路：线程池来帮忙，却好心办坏事

### 走过的弯路，你的套路

### ThreadLocal 是救火队长

### 悔不当初：回顾进化之路

### ThreadLocal 的第二种用法 part1

### ThreadLocal的第二种用法 part2

### ThreadLocal 的好处

### ThreadLocal 的原理

### ThreadLocal 的重要方法介绍

### 重要方法的源码分析

### 两种场景，殊途同归

### 收不回来的 value

### ThreadLocal 的空指针异常问题

### ThreadLocal 在 Spring 中的应用

## 不可不说的"锁"事【种类繁多，如何一一突破】

## atomic 包【一刻也不能分割】

### 什么是原子类、有什么作用

### 6 类原子类纵览

### AtomicInteger 案例演示

### Atomic 数组，每个都安全

### Atomic 引用在自旋锁的应用

### 把普通变量升级为原子变量

### 对比新旧工具的运行速度

### 剖析高速运转的原理

### 功能升级，不限于加操作

### 总结原子类

## CAS【不可中断的典范】

### 什么是 CAS

### CAS 的等价代码、使用案例

### CAS 的应用场景、源码分析

### CAS 的缺点和总结

## 以不变应万变【最便捷的并发安全之道】

### 什么是"不变性"？

### final 变量的赋值时机

### final 的注意点

### 栈封闭技术

### 面试题：真假美猴王

## ConcurrentHashMap 等并发集合【面试高频考点】

## 控制并发流程【做好线程之间的协调人】