

Java并发理论知识框架总结

多线程简介

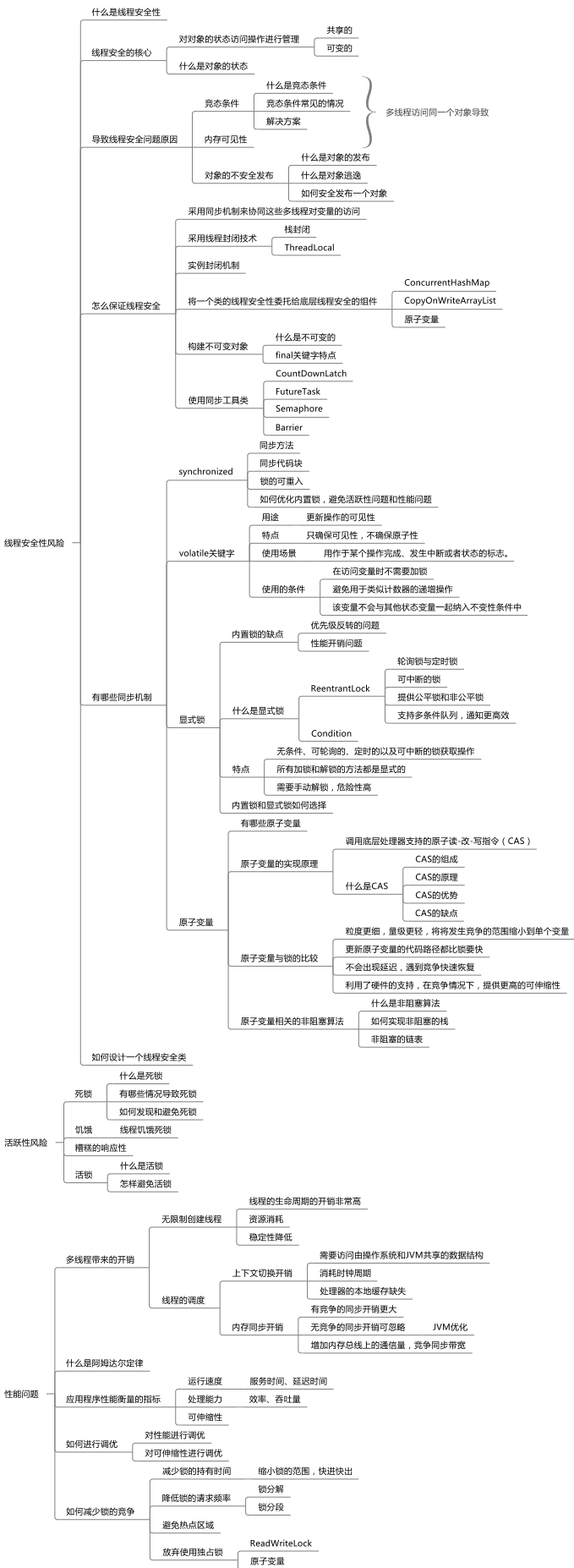
什么是线程
多线程的优点

Thread
Runnable
Callable
Future

怎样创建线程

线程的状态及转换

多线程带来的风险以及解决



线程池

什么是线程池
线程池的优点

Executors.newFixedThreadPool
Executors.newCachedThreadPool
Executors.newSingleThreadExecutor
Executors.newScheduledThreadPool
new ThreadPoolExecutor()自定义构建

如何创建线程池

- 如何正确设置线程池大小
- 线程池队列的用途及特性
- 如何选择线程池
- 线程池有哪些饱和策略

Executor框架介绍及特点

Runnable、Callable和Future在Executor中的角色及作用

Executor中任务取消和线程中断

- 哪些任务可以取消
- 什么是任务取消
- 任务取消的缺点
- 线程中断
 - 如何中断线程
 - 如何处理线程中断
 - interrupt()、isInterrupted()和interrupted()三个中断相关方法的特点
 - 传递异常
 - 恢复中断状态