零基础学 Java

1

初识线程

- 线程是什么
- 线程的执行
- 让线程睡一会儿
- 通过 debug 看线程

线程是什么

- 线程，英文名叫做 Thread，是 Java 程序执行的发动机。就是线程运行着我们的代码

- 一个程序就是操作系统的一个进程，英文名叫做 Process。一个进程下可以有很多线程

线程的执行

- 线程其实就是执行一个入口方法，执行完毕就结束了。比如我们之前写的程序，都是使用一个线程执行 main 方法，执行完毕后，线程就结束了

- 线程在执行方法的时候，每次遇到方法调用，都会给当前的线程栈增加一层。这一层里保存的，就是线程当前的执行状态，比如当前方法的局部变量的值，当前方法执行到哪里了等

- 所以线程栈里的每一条，都是方法已经开始执行但是还没有结束的方法。没有结束是因为它代码还没执行完，或者是在等待其调用的方法执行完

让线程睡一会儿

- 看例程：让线程睡一会儿

通过 debug 看线程

- 看例程：用 debug 断点功能，让程序停住。学习使用 Intellij 查看进程里有多少线程，每个线程的线程栈是什么
- 通过让方法执行结束，观察线程栈的变化

2

创建自己的线程

创建自己的线程

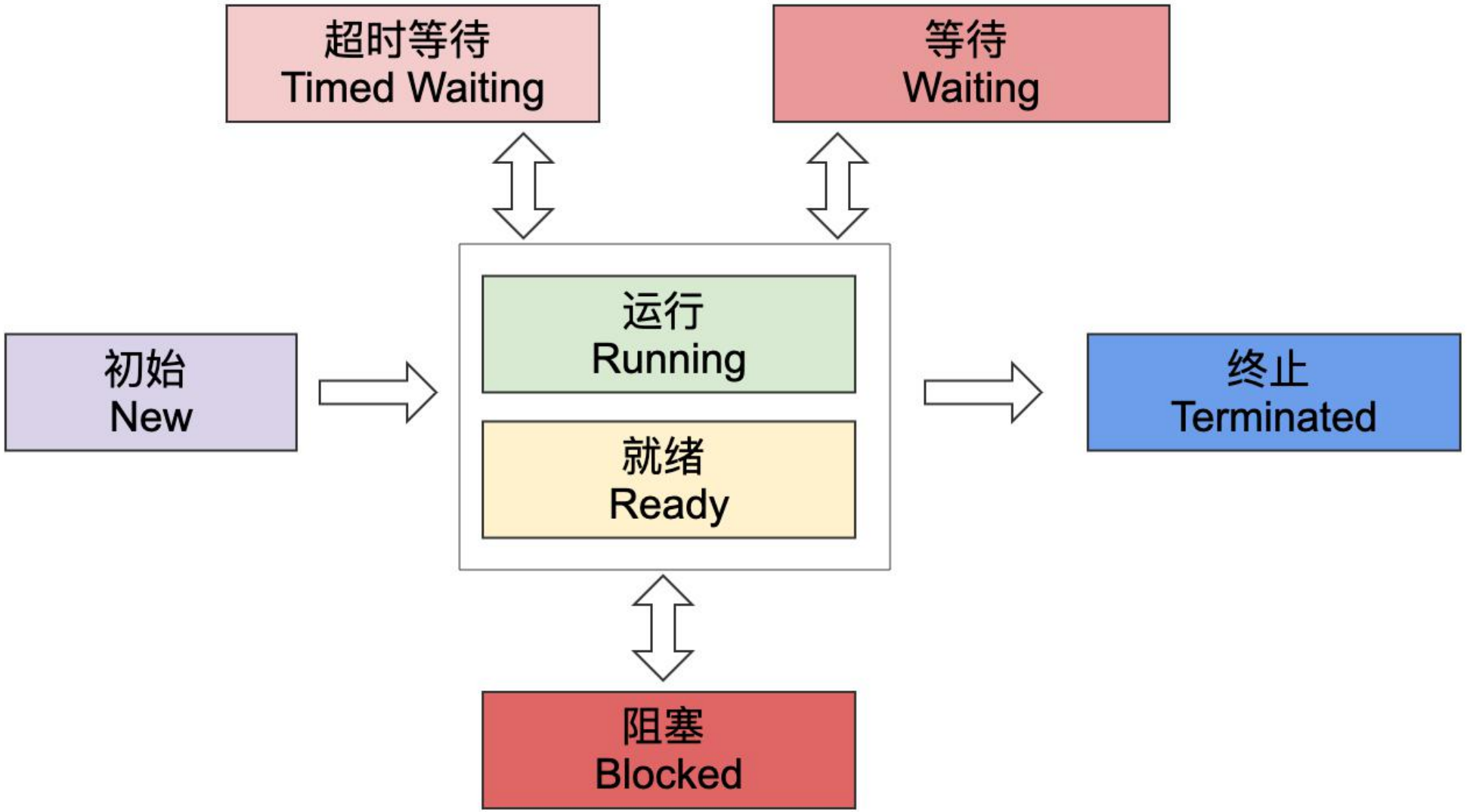
- 看例程：学习和理解线程
- 线程也是一个对象，执行完毕 Runnable 接口里的 run 方法，线程就结束了
- 当一个进程里所有的线程都执行结束了，一个进程就执行结束了
- 线程相当于是 CPU，它会从入口开始执行代码。一段代码可以被多个线程同时执行。可以通过 Thread.currentThread() 获取执行当前代码的线程
- 代码就好像曲谱，线程就好像乐器。乐器可以演奏曲谱，多个乐器可以演奏相同的或者不同的曲谱

3

再探线程

- Java 线程的状态
- 守护线程和优先级属性
- 线程的 interrupt 方法

Java 线程的状态


超时等待
Timed Waiting
等待
Waiting
初始
New
运行
Running
就绪
Ready
终止
Terminated
阻塞
Blocked

守护线程和优先级属性

- 守护线程：英文名叫做 daemon thread。如果一个进程里没有线程，或者线程都是守护线程，那么进程就结束了
- 看例程：只有守护线程，进程就结束了
- 看例程：可以设置线程的优先级，优先级的作用不能保证，这和线程的运行状态以及机器本身的运行状态有关。是不是守护线程都可以设置线程优先级

线程的 interrupt 方法

- 看例程：线程的 interrupt 无法真的像这个方法的名字那样让线程中断
- 看例程：线程的 stop 方法可以让线程结束，但是这会带来很大的隐患，会造成程序状态的错误，比如锁没有释放等。不要在生产的代码里调用这个方法

4

多线程：混乱开始了

多线程：混乱开始了

- 看例程：单线程版的修改数据，妥妥的
- 看例程：多线程版的修改数据，懵懵的
- 理解多线程版的结果：线程修改数据，人多手杂，一个线程在改，另一个线程也在改。读取当前值，修改为新的值，写入新的值这三个步骤并非是连续执行的，可能有别的线程的代码乱入。而且现代计算机的 CPU 都有缓存，让问题就更不可预测

5

同步控制之 synchronized

同步控制之 synchronized

- synchronized 关键字用来修饰成员方法，代表这个方法对于同一个对象来说，同一时间只允许一个线程执行，别的线程如果也调用这个实例的这个方法，就需要等待已经在执行这个方法的线程执行完毕，才能进入方法执行
- 看例程：使用 synchronized 解决问题
- 看例程：使用 synchronized 修饰静态方法
- 看例程：使用 synchronized 代码块

6

同步控制之 wait notify

同步控制之 wait notify

- 来自 Object 类里的方法
- 应用场景：正如名字所说，当多个线程的互动，需要等待和和被唤醒的时候，就可以考虑使用这个语法
- 看例程：学习 wait notify / notifyAll 语法
- 看例程：学习 lost notification
- 知识点 + 注意点：synchronized 不是公平锁

7

多线程经典模型：生产者消费者

多线程经典模型：生产者消费者

- 生产者消费者模型简介
- 生产者消费者重点：生产的任务不能被忽略或者多次消费
- 看例程：使用 List 和 wait notify 实现生产者消费者

多线程的思维转变

- 多线程第一课：程序随时可能停住，别的线程随时可能乱入任何代码

8

线程同步之 join

线程同步之 join

- 线程 join 方法的作用
- 看源码：join 方法的文档
- 看例程：学习 Thread 的 join 方法

9

死锁

死锁

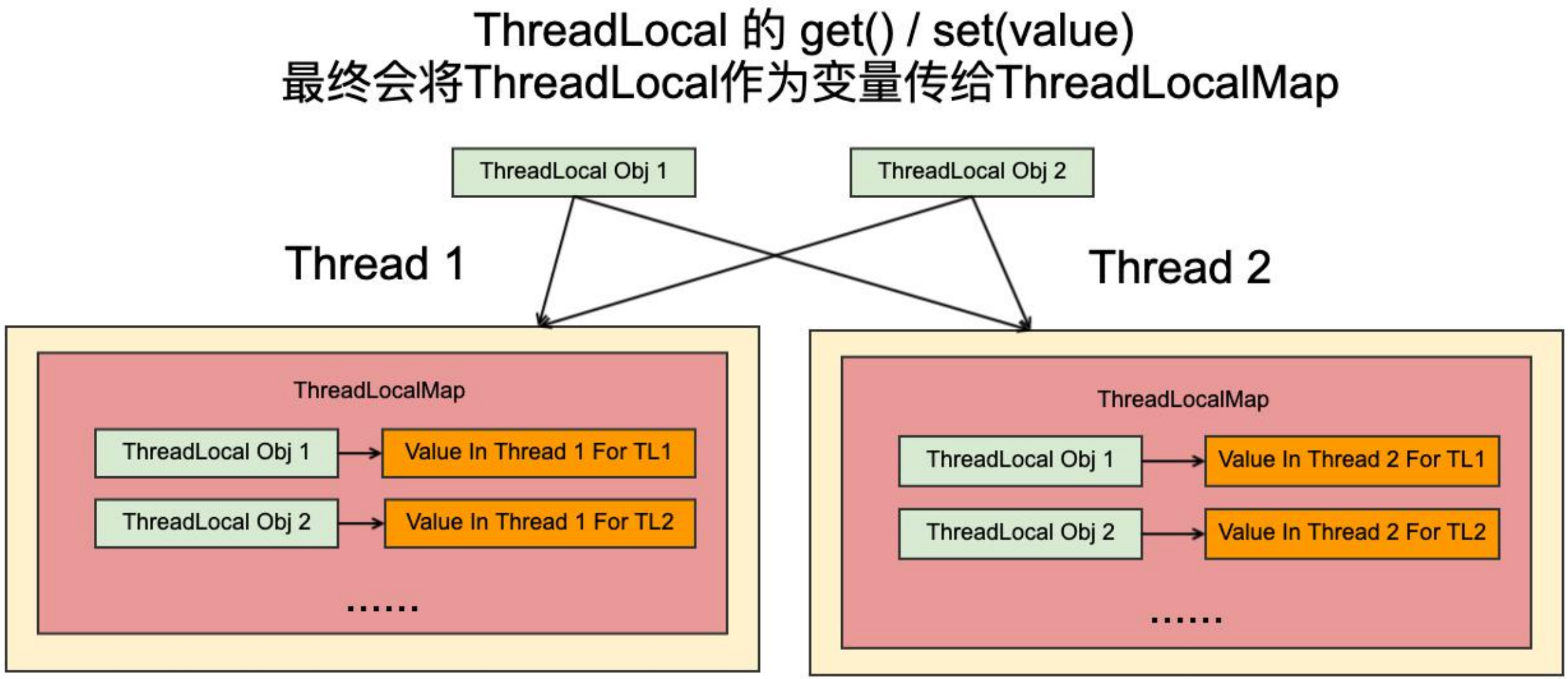
- 一种死锁形成的条件：在获取新的资源之前，没有释放之前获取的资源
- 看例程：模拟死锁形成的条件
- 工具：使用 jstack 查看死锁信息
- 如何避免死锁：按照顺序申请资源

10

ThreadLocal：线程专属的变量

- ThreadLocal 的原理
- ThreadLocal 的用法

ThreadLocal 的原理



ThreadLocal 的用法

- 看例程：学习使用 ThreadLocal 跟踪线程执行各个阶段的时间
- ThreadLocal 源码阅读提示：ThreadLocalMap 里处理 hash 冲突的机制不是像 HashMap 一样使用 List。它采用的是另一种经典的处理方式，沿着冲突的索引向后查找空闲的位置

11

定时任务

定时任务

- 看例程：学习 Date，Calendar 和 SimpleDateFormat
- 看例程：使用 Timer 执行定时任务

12

volatile 关键字

- JMM 和 指令重排
- volatile 关键字的作用

JMM 和 指令重排

- Java Memory Model：简称 JMM，翻译为 Java 内存模型。我们可以简单的认为是一套 happens-before 标准，规定了内存同步和缓存失效等节点，限制了指令重排

- 看例程：指令重排

volatile 关键字的作用

- 看例程：volatile 关键字的变量对指令重排的影响
- 看例程：volatile 关键字强制每次都从主存获取变量数据
- JMM 是 Java 的内涵之一。Java 字节码（Java Byte Code）使得 Java 在指令层面有了统一的标准。JMM 更让 Java 在执行优化层面也有了统一的标准。让各大厂商可以根据操作系统和硬件，在执行优化上放飞自我

13

同步大杀器：concurrent 包

- concurrent 包简介
- concurrent 包基本原理

concurrent 包简介

- concurrent 包的起源： 开始是由 Doug Lea 博士开发的。 Doug Lea 博士先是提供了一个独立于 JDK 的 concurrent 包，后来主导了 JSR 166，将这个软件包纳入到了 JDK 1.5 中。造福了广大 Java 开发者

- JSR 是 Java Specification Requests 的简称。是 Java 演进的标准。每个 Java 版本的新功能都是以 JSR 的形式推进的

concurrent 包基本原理

- 使用 CAS，避免内核调用。CAS 是 Compare And Swap 的缩写。CAS 命令是现代 CPU 都支持的一种指令。这个指令对一个数据的写操作，需要三个操作数：内存里的值的地址，旧的值，新的值。只有当 内存里的值 == 旧的值，内存里的值才会被设置为新的值。而且这个操作是原子操作，不会被 CPU 执行调度打断

- 看源码：park 和 unpark

- 理解 CAS 如何实现锁的功能和子旋

- 有了新的锁的机制，就可以在同步的道路上越走越宽了

14

concurrent 包中的 Atomic 类族

concurrent 包中的 Atomic 类族

- 每种基本数据类型对应的 Atomic 类，以及引用类型的 Atomic 类
- 查看 AtomicLong 的源代码，了解实现原理
- 看源码：AtomicLong 的线程安全性

15

concurrent 包中的锁

concurrent 包中的锁

- 看例程：学习 Lock 如何对标 synchronized
- 看例程：学习 Lock 如何对标 wait / notify 机制
- 看例程：学习 CountdownLatch 如何替代 Thread 的 join

16

concurrent 包中的数据结构

concurrent 包中的数据结构

- concurrent 包中的数据结构以高效支持线程安全为特色
- 看例程：学习 LinkedBlockingQueue
- 看例程：学习 ConcurrentHashMap

17

concurrent 包中的线程池

concurrent 包中的线程池

- 什么是线程池
- 看例程：学习线程池的使用

18

聊天室开张喽！

聊天室开张喽！

极客时间

功能描述

- 可以进入聊天室，指定用户名，检查用户名重名
- 服务器端消息转发
- 可以任意两人聊天
- 聊天内容仅限文本
- 可以进行查看在线用户列表等服务端功能

聊天室开张喽！

- 功能演示
- 看代码：介绍程序设计和关键点

聊天室开张喽！

极客时间

习题：

- 群发消息
- 发送文件
- 心跳检测，自动去除断开连接的客户端