

# 《图解Java多线程设计模式》 阅读计划

——图灵后端技术群阅读计划（第1期）

领读人：橙

## 本书特色

- 详细讲解多线程模式及相关Java语言功能
- 所有模式配有具体的 Java 语言示例程序
- 模式名称的英文读法、含义及中文表述
- 除第 13 章外，每章末尾都配有练习题
- 附录D 总结了一些线程相关的主要API 文档
- java.util.concurrent 包及主要类的用法介绍

## 适合读者

- 有一定工作经验并且编程语言以Java为主的编程人员
- 对Java的多线程感兴趣的工作人员
- 在面对解决单线程同步阻塞，寻找新的设计模式的中级开发人员

总阅读时间长度（预估）：3周

每天用时（共3小时）：阅读2小时，练习题1小时

答疑时间安排：每周一次，每周日晚图灵后端技术群20:00—22:00

图灵社区本书网址：<http://www.ituring.com.cn/book/1812>

图灵阅读计划网址：<https://github.com/BetterTuring/turingWeChatGroups>

## 阅读建议

- ①配置好Java相关的环境，推荐开发JDK为[JDK 1.8](#)
- ②配置好自己的集成开发环境（IDE），推荐使用[IntelliJ IDEA](#)的社区版本
- ③保持良好的阅读习惯，坚持边做边练

# 阅读规划

务必在阅读主要章节前，先阅读[关于UML]相关内容。

## 第一部分（序章1，2以及第1—4章）

阅读时长：一周

### 序章1与序章2

- 序章1介绍了Java线程的相关基础知识。了解Java在多线程编程所使用的相关工具。
- 序章2介绍了多线程程序的评价标准。涉及到四个指标：安全性、生存性，可复用性以及性能。

“

序章1与序章2的页面范围：[1-34]，共计34页  
安排时间：1 DAY（当前周：第1周）

#### 说明

- 此为多线程编程的一些**基本工具**以及**概念**，务必夯实基础，以免在后续编程过程中产生不必要的误解以及麻烦。
- 主要讲了关于线程的两种使用方式，以及一些可能会混淆的概念，在练习题中有所体现。

记住一句古话：磨刀不误砍柴工。

### 第1章（Single Threaded Execution模式）

- 第1章为单例线程执行设计模式，通过设计限制，以保证同一时间仅有一个线程可以运行。

“

第1章的页面范围：[35-60]，共计25页  
安排时间：1 DAY（当前周：第1周）

#### 说明

- 这是我们所要学习的**第一个**设计模式，同时，这也是**多线程程序设计的基础**。在日后的学习过程中，我们不仅要做到知其然，更要做到知其所以然。为何要使用单线程执行模式，所适用的场景又是如何的呢？例如银行的区块操作，即便有多个人登录你的账户，也只能排队依次按照顺序一个个地来取款。

## 第2章 (Immutable模式)

- 第2章为不改变模式，旨在不执行耗时的互斥操作下，不影响程序的安全性，一旦创建，即不可改变，无论多少次访问也不会影响到它以至于发生变化，以此来提供计算机性能。

“

第2章的页面范围: [61-80], 共计19页  
安排时间: 1 DAY (当前周: 第1周)

### 说明

- 事实上，并非所有的操作都需要提供互斥操作，例如：String就是一个一次生成，再也不会被改变的类。在实际生产过程中，存在着需要保存实例状态却不允许其他人改变它的情况，采用此模式，即可避免耗时的互斥操作，来提高计算机性能。
- 例如我们浏览的淘宝页面，我们并不能对它进行修改，他们是由淘宝后台统一来进行修改的，我们只有浏览的权限，但是当别人与我们查看相同内容的时候，并不会因为我们在查看而导致别人无法查看网页相关内容。

## 第3章 (Guarded Suspension模式)

- 第3章为守护暂停模式，当我们收到线程通知时，让它暂时等待，等到合适的时机再和他通讯，通过等待来保证线程的安全性。

“

第3章的页面范围: [81-98], 共计17页  
安排时间: 1 DAY (当前周: 第1周)

### 说明

- 这一章所囊括的内容较多，稍稍复杂了一些，多了一些名词，需要读者用心阅读，理解用意，配合场景去了解这样的设计模式。

## 第4章（Balking模式）

- 第4章为停止并返回模式，结合本章一开始的场景去理解此设计模式，你需要呼叫服务员，当服务员A响应并来到你身边时，服务员B正准备来，可当服务员B看到A已经在的时候，服务员B就停止过来了。AB服务员就如同两个响应线程一样。

“

第4章的页面范围：[99-114]，共计15页  
安排时间：1 DAY（当前周：第1周）

### 说明

- 多线程设计模式并非泛泛而谈，除了服务员的例子，我们身边还有很多诸如此类的例子，例如我们常见的滴滴打车，不就是如此吗？当一个用户约到了一辆车，即便其他的车子也接到了单子，但当用户已经约到了车，其他的响应就会自动丢弃了。

## 答疑安排（1 Day）

- 针对一些练习进行解答，并且对设计模式的场景提出思考。与大家进行交流。
- 

## 第二部分（第5—9章）

阅读时长：一周

## 第5章（Producer-Consumer模式）

- 第5章为生产者-消费者模式，在生产者与消费者之间加入一个“桥梁角色”，用于处理生产和消费之间处理速度的差异。

“

第5章的页面范围：[115-140]，共计26页  
安排时间：1 DAY（当前周：第2周）

### 说明

- 这个模式在我们日常生活中可谓非常常见。例如买早餐豆浆举例，通常会有一位师傅在后面灌豆浆，有一个人负责卖豆浆，客人每隔几分钟来一波，为了保证及时响应顾客，

需要提前准备好一些豆浆以便急客带走，待生意闲下来的时候也可以往货柜上补满豆浆。因为灌豆浆的速度比不上购买豆浆的速度，故我们先准备好一批以应付高峰，待闲时再补充储备。

- 线程亦是如此，准备好一批线程在内存中等待，当需要时使用，并且不停地补充，待闲时又补充到一定数量的线程以待消费。

## 第6章（Read-Write Lock模式）

- 第6章内容为读写时锁模式，即读时不可写，写时不可读。

“

第6章的页面范围：[141-162]，共计22页  
安排时间：1 DAY（当前周：第2周）

### 说明

- 例如书上的例子，老师在黑板上写了很多的内容，学生们此时正在记录黑板上的内容，此时老师想在黑板上写新的内容，便想把黑板擦了。由于还有学生正在摘抄黑板上的内容，所以学生就告诉老师先不要擦，等到学生都摘抄完了，老师才开始擦黑板并书写新的内容。
- 以上实例即为当所有读线程终结，无人再读时才进行写入操作，保证了事物的一致性。
- 同样它还有另外一个名字叫做多读单写模式。多个人可读，一个人写，有些类似于**一主多从**的概念。

## 第7章（Thread-Per-Message模式）

- 第7章内容为任务委托模式，分为委托端与执行端两个部分。

“

第7章的页面范围：[163-186]，共计24页  
安排时间：1 DAY（当前周：第2周）

### 说明

- 举个现实的例子。老板希望知道全国每个省的销售情况，他可以自己一个人去跑所有的省（如果他愿意……）一般这样的情况不会发生，因为太过于繁琐也太过于浪费时间了。老板会委托很多人同时去各个省份去调查，这样其他的人就可以并发地去处理，节约了大量的时间。

- 线程也是如此，如果单线程同步向数据库发起请求，假设一个请求需要5秒，依次发起8个请求的话，那用户就需要等待40秒，但如何同时发起8次请求（创建线程的时间忽略不计），那么忽略其他因素，仅仅需要5秒即可响应客户了。

## 第8章（Worker Thread模式）

- 第8章内容为工人线程模式，这个跟第5章有些类似的地方，但是也有一些不同的地方。虽然都是准备好一批东西。但是第5章是消费掉，不可复用；而工人线程模式可复用。

“

第8章的页面范围：[187-210]，共计24页  
安排时间：1 DAY（当前周：第2周）

### 说明

- 假设北京的胡同里有很多的车夫，假设有100个，这时候来了50个客人，他们马上就可以享受到服务，而另外50个会闲置继续等待客人的到来。也许有一天，客人并发的数量超过了100，那么拉车公司可能考虑聘用更多的车夫。
- 线程池的意义即在于此，但是足够多的线程也可能撑爆你的JVM，所以当你使用线程池的时候，请了解你的环境是怎样的。

## 第9章（Future模式）

- 第9章内容为未来模式。与任务委托模式类似，但不同的是，这个明确了主线程的参与以及目的不同，是委托了别人完成，但重点不在于此，而在于我们不需要等待他人完成才进行下一步了。

“

第9章的页面范围：[211-230]，共计20页  
安排时间：1 DAY（当前周：第2周）

### 说明

- 倘若我们需要购买一个蛋糕，你会到了蛋糕店傻傻地等待蛋糕做出来吗？
- 这时候，收银员给了你一张单据，告诉你在未来的一段时间内，随时可以过来提取（约定好了时间，除非你希望那个蛋糕一直放在他们冰箱里）。这时候，你便可以离开蛋糕店，去买些红酒以及牛排，准备好今日的生日晚餐，待到黄昏时分，蛋糕也做完了，你带着单据开开心心地提走了蛋糕，过了一个愉快的夜晚。

- 在面对一些IO密集型等待时间较长的操作的时候，我们可以先让别人去做，我们等到合适的时机再去取即可，而不需要等待它完成才能进行下一步操作。

## 答疑安排 (1 Day)

---

## 第三部分 (第9—13章)

阅读时长：一周

### 第10章 (Two-Phase Termination模式)

- 第10章内容为分两阶段终止模式。即当我们通知线程需要终止的时候，它会依据情况做一些预处理操作。
- 套用书中的内容，主要有三个要点：安全地终止进程（安全性），必定会进行终止处理（生存性），发出终止请求后尽快进行终止处理（响应性）。

“

第10章的页面范围：[231-262]，共计32页  
安排时间：1 DAY（当前周：第3周）

#### 说明

- 在此设计模式下，倘若我们希望结束线程，则可以执行一些我们所希望的操作，例如回滚，配合第4章的停止并返回模式是不是很好呢？假设我们已经发出请求撤回消息，其他的线程不是直接中断请求（那样太不优雅），而是先给一个抱歉，并且撤回了订单，然后再终止自己。是不是更加安全可靠了呢？
- 而当已经通过的订单，我们希望终止，它就会在完成订单之后才终止此进程。这样做，是不是既优雅又可靠，比起粗暴的强制退出可好多了？

### 第11章 (Thread-Specific Storage模式)

- 第11章内容为线程特定空间模式，即使在一个线程池中，每一个线程都有自己对应的一块特定的内存空间。

“

第11章的页面范围：[263-282]，共计20页  
安排时间：1 DAY（当前周：第3周）

## 说明

- 例如打印日志，若为系统日志则用默认线程打印至控制台，若为错误信息则用2线程打印至本地文件，若为数据库信息则打印至数据库。每一个线程执行不同的操作，并且拥有不同的职能。

## 第12章 (Active Object模式)

- 第12章内容为主动模式，其中糅合了第5章生产者消费者模式，第7章任务委托模式，第9章未来模式。

“

第12章的页面范围: [283-320], 共计38页  
安排时间: 1 DAY (当前周: 第3周)

- 可以综合起来想：我们开了一家早餐店公司，我是老板，雇佣了很多线程去工作，有些在收钱，有些在卖豆浆，有些在进货。而我只需要知道开支是多少收入是多少，以及当线程不足的时候及时补充线程即可。把这个整体贯穿起来。
- 这一章因为糅合了其他的设计模式，建议先阅读相关章节，再阅读本章节。

## 第13章 (总结)

- 第13章内容为总结。讲述了模式以及模式语言。如果将模式看作是针对一个问题的解决方案，那么模式语言就是针对某个领域中的问题集的解决方案集合。不管是在哪个领域，只要解决方案能很好地描述出技巧、心得、要领、提示等，就是模式语言。

“

第12章的页面范围: [321-335], 共计15页  
安排时间: 2 DAY (当前周: 第3周)

## 说明

- 通过此章的阅读，我们应该对设计模式以及模式语言的关系有了初步的了解，阅读模式语言可以理解该领域的问题集与解决方案集，从中选择出可以解决自己遇到的问题的模式，然后运用它们。

## 答疑安排 (1 Day)