

# 课程学习须知

## 并发编程课程大纲

- 并发编程的发展以及价值 (5月15号)
- 并发编程带来的挑战之同步锁 (5月16号)
- 并发编程带来的挑战之可见性 (5月19号)
- 并发安全性之Lock锁及原理分析 (5月22号)
- 线程阻塞唤醒wait,notify以及condition,死锁等原理分析 (5月23号)
- J.U.C并发工具集场景及原理分析 (5月26号)
- 随便聊聊ThreadLocal&ForkJoin (5月29号)
- 深度剖析阻塞队列的设计原理及实现 (5月30号)
- **并发安全的集合ConcurrentHashMap (6月2号)**
- **站在架构的角度思考线程池的设计和原理 (6月5号)**
- **Java8新的异步编程方式 (6月6号)**

注意：随着课程的推进，有可能会增加课时来讲解，所以如果遇到课程大纲的变化，我会及时通知！

## 新同学注意事项

并发编程是一个完整的体系，到目前为止已经讲了9节课了，如果前面部分内容没有听的同学，一定要记得补一下，不然越到后面越听不懂。

另外，有些同学基础比较薄弱点，对并发这块完全不熟悉，我强烈建议这些同学至少听2遍以上。因为并发编程整个体系是偏基础和底层的，和我们平时开发用的应用框架的理解完全不一样。

这就需要大家逐步去转变一些学习思维，从而更好的理解。

有部分学员反馈有问题找不到我，记得加我微信 **mic4096**，qq目前登录较少，有些消息回应不及时。

## 本阶段课程变化

- 新增了部分内容，如异步编程、ForkJoin等。
- 内容更加细致。
- 讲课节奏会稍微放缓，帮助大家更好的吸收。

## 本阶段内容的学习目标

并发编程是很多大公司面试重点考察的范围，据部分拿到高薪Offer的学员反馈，大部分面试中，并发编程很好的帮助他们拿到Offer。

所以今年，对于并发编程的内容做了调整，更加细致和全面。

大家学完之后要达到的目标。

- 很好的理解线程的本质。
- 能够灵活运用线程。
- 从原理层面理解并发编程。
- **在并发编程领域击败99%的程序员**

# 本周预习资料

本周的三次可，应该会讲以下内容，condition部分因为没讲完成，所以会放在本周三讲剩下部分。

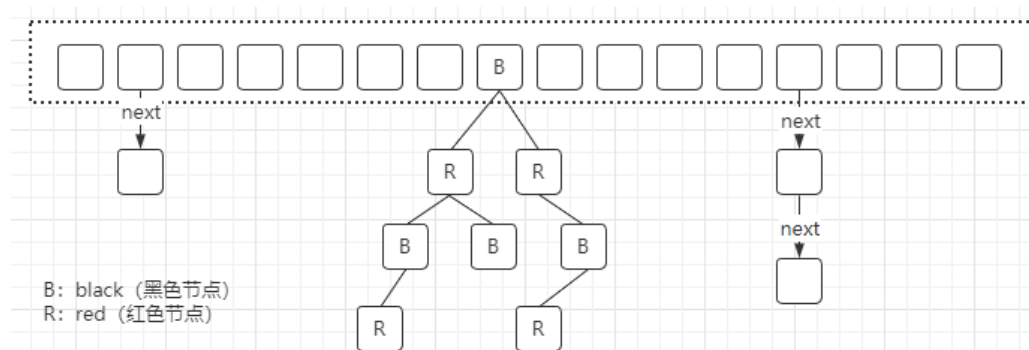
- 线程阻塞唤醒wait,notify以及condition,死锁等原理分析（5月23号）
- J.U.C并发工具集场景及原理分析（5月26号）
- ThreadLocal&ForkJoin（5月29号）
- 深度剖析阻塞队列的设计原理及实现（5月30号）

## 并发安全的集合ConcurrentHashMap（6月2号）

课前去看一下ConcurrentHashMap的源码，尝试分析，把不懂的地方标记出来

- ConcurrentHashMap的基本使用
- 正确理解ConcurrentHashMap的安全性
- ConcurrentHashMap的存储结构说明
- ConcurrentHashMap源码分析
  - 数组初始化
  - 数据扩容
  - 数据迁移
  - 分段锁设计
- 红黑树分析

在JDK1.8中，ConcurrentHashMap采用数组+链表+红黑树的方式来实现数据存储，数据结构如下图所示。



## 站在架构的角度思考线程池的设计和原理（6月5号）


线程池是用来实现线程复用的技术，如何实现线程复用，思考下线程池的设计吧？

- 线程池的优势
- Java中提供的线程池
- 思考线程池的设计
- 分析线程池的实现原理
- 线程池源码分析

## Java8新的异步编程方式（6月6号）

CompletableFuture是Java8中提供了一种异步编程的api，它提供了非常强大的异步回调通知机制。

- 了解Future/Callable
- Java8新特性之CompletableFuture
- CompletableFuture的使用

- 
- CompletableFuture综合实战
  - 原理及源码分析。

## 预习须知

---

同学们，预习很重要，但是预习不是给大家一堆资料去看，而是自己先去根据我要讲的内容查询一下相关资料，然后在查询过程中会产生很多疑问，而我在课堂上，会帮助大家解决这些疑问，这样的学习效果是更好的。

如果一味的去被动接受这些知识，没有思考。那么对大家来说，几遍是吸收了一些东西，但是最根本的学习能力还是不具备的。

技术的迭代速度非常快，我们需要有一些自我学习能力！！