**学习总结**

姓名\_\_董金伟\_ 专业\_计算机科学与技术\_ 班级\_\_\_2班\_ 日期\_\_2022/2/15\_

1. 当天所学习主要内容（做一总结）
2. 线程互斥：

互斥：间接相互制约。一个系统中的多个线程必然要共享某种系统资源。

共享资源----临界资源

同步块：

synchronized（对象）{

所有对共享资源操做的代码

不要把和共享资源操做无关的代码放同步块中，以提高效率。

}

每个线程中的同步对象要求是同一个对象。通常用this。

共享资源设为private。

工作原理：任意对象中都有一个 “锁”，有解锁状态，上锁状态两个状态。如果对象是解锁状态，则先上锁，然后执行同步代码块中的内容，同步块执行完后对对象解锁；如果对象是上锁状态，则处于阻塞状态。

同步方法：

访问权限 修饰符 返回类型 方法名(形参表){

对共享资源操做的代码。

}

修饰符：synchronized

非静态同步方法上锁的对象是this。

同一个对象在不同的线程中调用同步方法，某一时刻只能有一个线

程可以执行同步方法

注意：静态方法可以是同步方法。对所在的类对应的Class对象上锁解锁操做。类对应的Class对象只有一个。

1. 线程同步：

同步：线程之间的合作。如：生产消费问题。

wait()：等待。解锁、进入阻塞状态。

notify()：唤醒某一个处于等待的线程。结束阻塞状态，进入可运行状态。

notifyAll()：唤醒所有处于等待的线程。结束阻塞状态，进入可运行状态。

注意：要在同步块或同步方法种调用该方法，同时调用该方法的对象要

求是同步对象。

面试题：

wait和sleep的区别？

等待时 wait 会释放锁，而 sleep 一直持有锁。

wait 通常被用于线程间交互，sleep 通常被用于暂停执行。

wait是Object类中的方法；sleep是Thread类中的方法；

执行wait后进入等待状态；执行sleep后进入休眠状态；

子线程运行执行 10 次后，主线程再运行 5 次。这样交替执行三遍

1. 学习中所存在的问题及解决办法

面试题：子线程运行执行 10 次后，主线程再运行 5 次。这样交替执行三遍

文本

描述已自动生成

题目需要两个线程分别执行，而我写成了一个线程，不符合题意。

3．学习收获及建议

学习了如何用同步来实现对线程的操作，昨天没有学习同步时发现。买票时，两个线程会同时买到一张票，这是不符合预期的，加入同步后即可解决线程互斥问题。

同步代码块：synchronized(this){ 代码块 }

同步方法：synchronized int 方法名() { 代码块 }

以上两种方法可以解决这个问题。

除了线程之间的互斥，常见的还有线程的合作，例如生产者消费者问题。

其中需要使用wait(), notify()两个方法来实现线程的等待和唤醒。

使用wait()时可能会遇到死锁。解决死锁的方法有，相同的顺序执行wait(),notify()。

练习例题时发现自己并没有理解这个知识点，面对面试题不知道改怎么写。