

多线程

一. 选择题

1. 下列说法中错误的一项是 (A)
A. 线程就是程序 B. 线程是一个程序的单个执行流
B. 多线程是指一个程序的多个执行流 D. 多线程用于实现并发
2. 下列哪个操作不能使线程从等待阻塞状态进入对象阻塞状态 (D)
A. 等待阻塞状态下的线程被 notify() 唤
B. 等待阻塞状态下的线程被 interrupt() 中断
C. 等待时间到
D. 等待阻塞状态下的线程调用 wait() 方法
3. 下列哪个方法可以使线程从运行状态进入其他阻塞状态 (A)
A. sleep B. wait C. yield D. start
4. 下列说法中错误的一项是 (D)
A. 一个线程是一个 Thread 类的实例
B. 线程从传递给线程的 Runnable 实例 run() 方法开始执行
C. 线程操作的数据来自 Runnable 实例
D. 新建的线程调用 start() 方法就能立即进入运行状态
5. 下列关于 Thread 类提供的线程控制方法的说法中，错误的一项是 (D)
A. 在线程 A 中执行线程 B 的 join() 方法，则线程 A 等待直到 B 执行完成
B. 线程 A 通过调用 interrupt() 方法来中断其阻塞状态
C. 若线程 A 调用方法 isAlive() 返回值为 true，则说明 A 正在执行中
D. currentThread() 方法返回当前线程的引用
6. 下列说法中，错误的一项是 ()
A. 对象锁在 synchronized() 语句执行完之后由持有它的线程返还
B. 对象锁在 synchronized() 语句中出现异常时由持有它的线程返还
C. 当持有锁的线程调用了该对象的 wait() 方法时，线程将释放其持有的锁
D. 当持有锁的线程调用了该对象的构造方法时，线程将释放其持有的锁
7. 下面的哪一个关键字通常用来对对象的加锁，从而使得对对象的访问是排他的 A
A. srialize B transient C synchronized D static

二. 填空题

1. 在操作系统中，被称做轻型的进程是 线程
2. 多线程程序设计的含义是可以将一个程序任务分成几个并行的 任务
3. 在 Java 程序中，run() 方法的实现有两种方式：实现 Runnable 接口 和 继承 Thread 类
4. 多个线程并发执行时，各个线程中语句的执行顺序是 确定 的，但是线程之间的相对执行顺序是 不确定 的
6. Java 中的对象锁是一种独占的 排他 锁
7. 程序中可能出现一种情况：多个线程互相等待对方持有的锁，而在得到对方的锁之前都不会释放自己的锁，这就是 死锁
8. 线程的优先级是在 Thread 类的常数 MIN_PRIORITY 和 MAX_PRIORITY 之间的一个值
9. 处于新建状态的线程可以使用的控制方法是 start() 和 stop()。
10. 一个进程可以包含多个 线程

三．简答题

1. 说明几个概念：对象锁、sleep、wait、notify、notifyAll

对象锁：每个对象都有对象锁，当某个线程进入用 `synchronized` 定义方法时，就获得了当前类对象（方法所属的类的对象）的锁，其他线程就不能再获得该锁，直到线程释放该锁。一个类对象有一个对象锁，多个同步方法公用一个对象锁。

`sleep`：是当前线程处于休眠状态，将对 `cpu` 的控制权交给其他线程，但不释放当前对象的锁，其他线程可以获得 `cpu`，但不能进入当前线程控制的对象锁。

`wait`：与 `sleep` 的区别是，它会释放对象锁，其他线程可以进入该同步块。

`notify`：进入当前同步块的线程调用 `wait` 方法变成休眠状态的线程，`notify` 将该线程变成等待状态。

`notifyAll`：进入当前同步块的线程调用 `wait` 方法变成休眠状态，`notifyAll` 使所有休眠线程变成等待状态。

一个类中有多个同步方法，进入任意一个同步块都将获得该对象锁，由于 `wait` 方法使进入同步块的线程休眠，`notify` 只能唤醒所有被休眠线程中的一个，`notifyAll` 可以唤醒所有休眠线程。通常线程的通信通过 `wait`，`notifyAll` 和 `wait` 条件来实现。

2. java 中有几种方法可以实现一个线程？用什么关键字修饰同步方法？S

有两种实现方法，分别是继承 `Thread` 类与实现 `Runnable` 接口
用 `synchronized` 关键字修饰同步方法

3. sleep() 和 wait() 有什么区别？

答：`sleep` 是线程类（`Thread`）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 `sleep` 不会释放对象锁。`wait` 是 `Object` 类的方法，对此对象调用 `wait` 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 `notify` 方法（或 `notifyAll`）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

4. 同步和异步有何异同，在什么情况下分别使用他们？举例说明。

答： 如果数据将在线程间共享。例如正在写的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了， 那么这些数据就是共享数据， 必须进行同步存取。

当应用程序在对象上调用了一个需要花费很长时间来执行的方法， 并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

5. 启动一个线程是用 `run()` 还是 `start()`?

答： 启动一个线程是调用 `start()` 方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。`run()` 方法可以产生必须退出的标志来停止一个线程。

6. 多线程有几种实现方法， 都是什么？同步有几种实现方法， 都是什么？

答： 多线程有两种实现方法，分别是继承 `Thread` 类与实现 `Runnable` 接口
同步的实现方面有两种，分别是 `synchronized`, `wait` 与 `notify`

7. 线程的基本状态以及状态之间的关系

Java 中的线程有四种状态分别是：运行、就绪、挂起、结束