**并发与并行的区别：**

并发：多个任务在同一时间间隔执行

并行：多个任务在同一时间执行

**线程：**轻量的进程，是程序执行流的最小单元，有独立的程序计数器（下一条指令的地址）和执行堆栈，与其他线程共享进程的存储空间。

**多线程：**实现多个线程并发执行的技术。（是提高[资源](https://baike.baidu.com/item/%E8%B5%84%E6%BA%90)使用效率来提高系统的效率）

**线程池：**多线程处理形式，实现线程的复用

1）newSingleThreadExecutor

创建一个单线程的线程池。这个线程池只有一个线程在工作，单线程串行执行所有任务。如果线程因为异常结束，那么会有一个新的线程来替代它。此线程池保证所有任务的执行顺序按照任务的提交顺序执行。

2）newFixedThreadPool

创建固定大小的线程池。每次提交一个任务就创建一个线程，线程池的大小一旦达到最大值就会保持不变，如果某个线程因为执行异常而结束，那么线程池会补充一个新线程。

3）newCachedThreadPool

创建一个可缓存的线程池。会回收部分空闲（60秒不执行任务）的线程，当任务数增加时，可以添加新线程来处理任务。此线程池不会对线程池大小做限制，线程池大小完全依赖于操作系统（或者说JVM）能够创建的最大线程大小。

4）newScheduledThreadPool

创建一个大小无限的线程池。此线程池支持定时以及周期性执行任务的需求。

**守护线程：**

用户线程即运行在前台的线程，而守护线程是运行在后台的线程。守护线程作用是为其他前台线程的运行提供便利服务，而且仅在普通、非守护线程仍然运行时才需要，守护线程和用户线程的没本质的区别：唯一的不同之处就在于虚拟机的离开：如果用户线程已经全部退出运行了，只剩下守护线程存在了，虚拟机也就退出了

将线程转换为守护线程可以通过调用Thread对象的setDaemon(true)方法来实现。

(1) thread.setDaemon(true)必须在thread.start()之前设置，否则会跑出一个IllegalThreadStateException异常。你不能把正在运行的常规线程设置为守护线程。

(2) 在Daemon线程中产生的新线程也是Daemon的。

(3) 守护线程应该永远不去访问固有资源，如文件、数据库，因为它会在任何时候甚至在一个操作的中间发生中断。

**ThreadLocal：**很多地方叫做线程本地变量（也有些地方叫做线程本地存储）。ThreadLocal为变量在每个线程中都创建了一个副本，那么每个线程可以访问自己内部的副本变量。

**Synchronized:**一个线程访问object的一个synchronized(this)同步代码块时，它就获得了这个object的对象锁。结果，其它线程对该object对象所有同步代码部　  分的访问都被暂时阻塞。

**线程间通信：**多个线程来共同有规律地执行完一件任务，且避免对共享变量的争夺。

1. 等待唤醒机制：wait()、notify()
2. 共享内存机制：多个线程需要访问同一个共享变量，谁拿到了锁（获得了访问权限），谁就可以执行。