# 线程池

1.为什么要创建线程池？

答：创建线程和消耗线程需要消耗大量的系统资源，再加上业务处理时长，会大大的降低工作效率，使用线程池可将创建好一定数量的线程放入线程池中，在需要时从线程池中取出。当线程任务执行完毕则放回线程池。

同时方便对线程进行管理，如果当前请求数量超过已有线程执行范围，比如说程序启动时创建了100个线程，现在有101个请求进来，每个请求分配一个线程去执行，那么线程就会不够用，这时线程池可以管理成多出来的一个请求就会进入排队等候，避免无休止的创建线程导致系统崩溃。

2.线程池的种类

1.newSingleThreadExecutor

创建一个单线化的线程池，他只会用唯一的工作线程来执行任务，保证所有的任务按照指定的顺序来执行。

2，newFixedThreadEXecutor

创建指定数量级线程的线程池，可控制线程的最大并发数，超出的线程会进入排队等候。

3.newCachedThreadExecutor

创建一个可缓存的线程池

如果线程池线程数量超出处理需要，则会回收空闲线程，若无可回收的，则新创建线程池。

4.newScheduleThreadExecutor

创建定时线程池，支持定时任务周期性的执行

定时任务使用ThreadPoolExecutor创建而不是Executor创建

3.扩展。。。

## Mysql和Oracle的区别

Oracle是完全支持事务的而Mysql支持innodb储存引擎行级锁情况下支持事务

# 乐观锁和悲观锁

1.乐观锁：顾名思义，很乐观，在拿数据的时候认为数据不会被进行修改，因此不会上锁，在提交更新的时候会判断在此期间数据有没有被修改。乐观锁适用于读多写少。

实现方式：1.数据版本实现，即数据加字段version来标识数据，每更新一次version+1，在我们提交更新数据的时候会拿到当前Version和第一次查询拿到的Version对比，如果不一致，则为过期数据。

2.悲观锁：每次取出的数据都认为他会被修改，

所以每次拿到数据的时候都会为改数据上锁，直到该数据锁释放，被下一操作获取锁进行操作。java synchronized 就是悲观锁的一种每次线程修改数据需要先获得锁。

InnoDB会根据隔离级别在需要的时候自动加锁（即两个事物均修改同一数据）

InnoDB也支持通过特定的语句进行显示锁定，这些语句不属于SQL规范：   
\* SELECT … LOCK IN SHARE MODE   
\* SELECT … FOR UPDATE

博客地址：<https://blog.csdn.net/weixin_30258027/article/details/96696637>