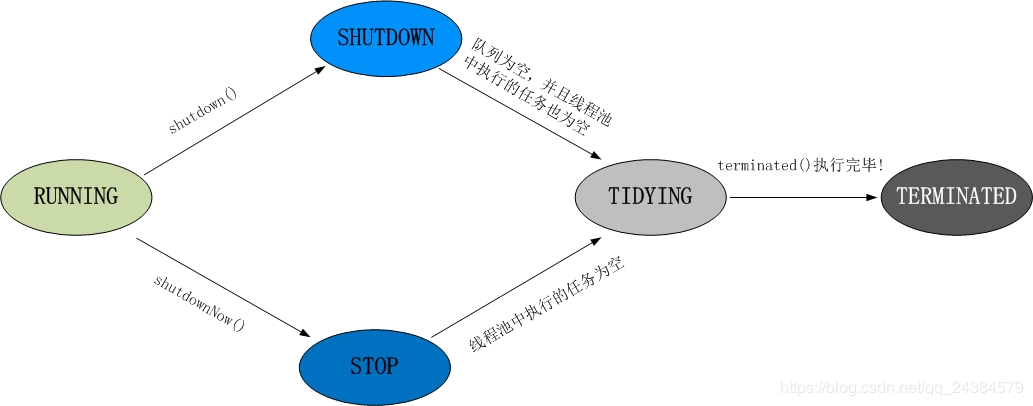
线程池的五种状态

线程池的5种状态：Running、ShutDown、Stop、Tidying、Terminated。

线程池各个状态切换框架图：



1.RUNNING

状态说明：线程池处在RUNNING状态时，能够接收新任务，以及对已添加的任务进行处理。

状态切换：线程池的初始化状态是RUNNING。换句话说，线程池被一旦被创建，就处于RUNNING状态，并且线程池中的任务数为0！

private final AtomicInteger ctl = new AtomicInteger(ctlOf(RUNNING, 0));

1

2.SHUTDOWN

状态说明：线程池处在SHUTDOWN状态时，不接收新任务，但能处理已添加的任务。

状态切换：调用线程池的shutdown()接口时，线程池由RUNNING -> SHUTDOWN。

3.STOP

状态说明：线程池处在STOP状态时，不接收新任务，不处理已添加的任务，并且会中断正在处理的任务。

状态切换：调用线程池的shutdownNow()接口时，线程池由(RUNNING or SHUTDOWN ) -> STOP。

4.TIDYING

状态说明：当所有的任务已终止，ctl记录的”任务数量”为0，线程池会变为TIDYING状态。当线程池变为TIDYING状态时，会执行钩子函数terminated()。terminated()在ThreadPoolExecutor类中是空的，若用户想在线程池变为TIDYING时，进行相应的处理；可以通过重载terminated()函数来实现。

状态切换：当线程池在SHUTDOWN状态下，阻塞队列为空并且线程池中执行的任务也为空时，就会由 SHUTDOWN -> TIDYING。

当线程池在STOP状态下，线程池中执行的任务为空时，就会由STOP -> TIDYING。

5.TERMINATED

状态说明：线程池彻底终止，就变成TERMINATED状态。

状态切换：线程池处在TIDYING状态时，执行完terminated()之后，就会由 TIDYING -> TERMINATED。