链滴

线程池的写法

作者：imyan

原文链接：https://ld246.com/article/1529543577247

来源网站：链滴

Java通过Executors提供四种线程池，分别为：

newCachedThreadPool创建一个可缓存线程池，如果线程池长度超过处理需要，可灵活回收空闲线，若无可回收，则新建线程。

newFixedThreadPool 创建一个定长线程池，可控制线程最大并发数，超出的线程会在队列中等待。

newScheduledThreadPool 创建一个定长线程池，支持定时及周期性任务执行。

newSingleThreadExecutor 创建一个单线程化的线程池，它只会用唯一的工作线程来执行任务，保所有任务按照指定顺序(FIFO, LIFO, 优先级)执行。

就你的例子来说，runnable 包业务逻辑或者业务逻辑包 runnable 没区别。现实场景下，需要仔细虑线程池的边界：

如果是对象聚合线程池，那么一般这个对象的方法会调用线程池；如果线程池是全局共享的某个通用步执行容器，

那么直接 executor.submit(()->{业务逻辑})就行，通常这里的业务逻辑最好是独立的、不依赖上下文。

四种线程池:

1.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需求。

2.newCachedThreadPool

创建一个可缓存的线程池。如果线程池的大小超过了处理任务所需要的线程，

那么就会回收部分空闲（60秒不执行任务）的线程，当任务数增加时，此线程池又可以智能的添加新程来处理任务。

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）能够创建的最大程大小。

```
public class T {
  public static ExecutorService executorService = Executors.newCachedThreadPool();
  public static ExecutorService executorService = Executors.newScheduledThreadPool(3);
  //new 单线程
  public static ExecutorService executorService =Executors.newSingleThreadExecutor();

  @Test
  public void testThread() {
      methodA();

      System.out.println("----------分割线----------");

      Runnable runnable = () -> {
          methodB();
      };

      executorService.submit(runnable);

  }

  public static void methodB() {
      System.out.println("business");
  }
```

```
    public static void methodA() {
        Runnable runnable = () -> {
            System.out.println("business1");
            // log.info("business");
  };
        executorService.submit(runnable);
    }
}
```