

面试官：“创建线程有哪几种常见的方式？”

1. 继承 Thread 类
2. 实现 Runnable 接口
3. 使用 Executor 框架
4. 使用 FutureTask

最简单的两种方式

1. 继承 Thread 类
2. 实现 Runnable 接口

比较实用的两种方式

3. 使用 Executor 框架

Executor 框架是 Java5 之后引入的，在 Java 5 之后，通过 Executor 来启动线程比使用 Thread 的 start 方法更好，除了更易管理，效率更好（用线程池实现，节约开销）外，还有关键的一点：有助于避免 this 逃逸问题。

补充：this 逃逸是指在构造函数返回之前其他线程就持有该对象的引用。调用尚未构造完全的对象的方法可能引发令人疑惑的错误。

Executor 框架不仅包括了线程池的管理，还提供了线程工厂、队列以及拒绝策略等，Executor 框架让并发编程变得更加简单。

为了能搞懂如何使用 Executor 框架创建

Executor 框架结构(主要由三大部分组成)

1) 任务(Runnable / Callable)

执行任务需要实现的 Runnable 接口 或 Callable 接口。Runnable 接口或 Callable 接口 实现类都可以被 ThreadPoolExecutor 或 ScheduledThreadPoolExecutor 执行。

2) 任务的执行(Executor)

如下图所示，包括任务执行机制的核心接口 Executor，以及继承自 Executor 接口的 ExecutorService 接口。ThreadPoolExecutor 和 ScheduledThreadPoolExecutor 这两个关键类实现了 ExecutorService 接口。

4. 使用 FutureTask