# 概述

## 简介

本文档主要记录学习JDK1.8中有关线程、线程池的学习笔记。

# 线程相关

## 创建线程方式

### 实现Runnable接口

具体实现参见Demo项目中的ThreadDemo，其调用方式如下：



图- 1实现Runnable接口创建线程

### 继承Thread类

具体实现参见Demo项目中的ThreadDemo，其调用方式如下：



图- 2继承Thread类创建线程

### 线程池获取

## 相关类

### Runnable

一个接口，只包含一个方法run，所有实现了该接口的类都必须实现run方法来定义自己的 业务逻辑。

重点关注FunctionalInterface（函数式接口）注解，如果一个Interface被FunctionalInterface注解标注，表明该接口中只能有一个抽象方法。如果有多个抽象方法（可以有多个方法，但只能有一个抽象方法），编译器直接检测到错误。如果接口中存在Default标注的默认实现方法，该方法不算作抽象方法，因此不会报错；或者接口中有static方法，因为static方法不算做抽象方法，也不会报错。

### Thread

#### 属性列表

表格 1列举了部分Thread类中属性，包括对象属性和类变量。

表格 1 Thread重要属性

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 类变量/对象属性 | 描述 |
| name | String | 对象属性 | 指定线程的名字，该属性被volatile修饰 |
| priority | int | 对象变量 | 描述线程的优先级，取值为[1,10]，CPU调度时会用到，优先级越高，获得时间片的概率越大。默认值为5 |
| daemon | boolean | 对象变量 | 标记线程是否为守护线程 |
| tid | long | 对象变量 | 用于表示线程的ID，每一个线程都有一个唯一的ID，其值通过threadSeqNumber赋值所得 |
| threadInitNumber | int | 类变量 | 如果在创建Thread对象时没有传递name属性，则通过该变量设置name；如果传递了name属性，该变量值不会改变 |
| threadSeqNumber | long | 类变量 | 每创建一个Thread对象将自增1，并且只有在创建对象时该变量才会自增，其他任何时候都不会自增，因此threadSeqNumber的值表明当前虚拟机已经创建过的线程数（并不是存活的线程数） |
| threadStatus | int | 对象变量 | java线程的状态，详见1.3。初始为0，并且该变量被volatile修饰 |

## java线程状态

参考：<https://blog.csdn.net/pange1991/article/details/53860651/>

参考：Thread类定义中的State枚举类

### 状态列表

Java线程一共有6个状态，分别如下表所示。这6中状态只针对Java线程，而不是操作系统线程的状态。各个状态之间的转换详见1.3.2。

|  |  |
| --- | --- |
| 状态 | 描述 |
| New | 当一个线程被创建当还未调用start方法时，此时线程处于New状态 |
| Runnable | 此状态包括两种情况，一种是Ready，一种是Running。当线程的start方法被调用后，线程将位于可运行线程池中等待被调度，获取CPU，此时处于Ready状态；当线程获取到CPU时，线程处于Running状态 |
| Blocked | 线程如果未获取到锁将被阻塞，此时线程将处于Blocked状态 |
| Waiting | 线程在执行过程中如果调用了Object.wait（）；Thread.join（）；LockSupport.park（）方法时，线程将进入Waiting状态。 |
| Timed\_Waiting | 线程在执行过程中如果调用了Thread.sleep()；Object.wait(long)；Thread.join（long）;LockSupport. parkNanos();LockSupport.parkUntil()方法时，线程将进入Timed\_Waiting状态 |
| Terminated | 线程完成执行之后将进入种终止状态 |

### 状态机

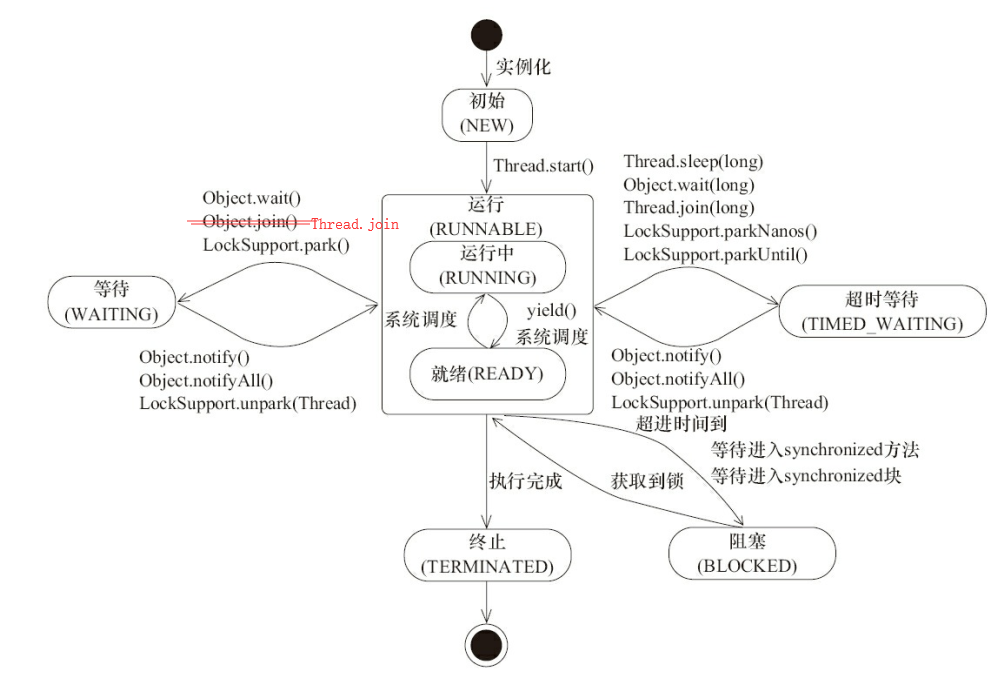


图- 3 Java线程状态机

# 线程池

# 参考

《Java并发编程的艺术》