

# 深入浅出线程Thread类的start()方法和run()方法

程序员欧阳

关注

赞赏支持



## 深入浅出线程Thread类的start()方法和run()方法

程序员欧阳

关注

 0.264

2018.05.26 17:21:47

字数 1,461

阅读 3,321

### 一、初识

java的线程是通过 `java.lang.Thread` 类来实现的。VM启动时会有有一个由主方法所定义的线程。可以通过创建Thread的实例来创建新的线程。每个线程都是通过某个特定Thread对象所对应的方法 `run()` 来完成其操作的，方法run()称为线程体。通过调用Thread类的 `start()` 方法来启动一个线程。

在Java当中，线程通常都有五种状态，创建、就绪、运行、阻塞和死亡。

第一是**创建状态**。在生成线程对象，并没有调用该对象的start方法，这是线程处于创建状态。

第二是**就绪状态**。当调用了线程对象的start方法之后，该线程就进入了就绪状态，但是此时线程调度程序还没有把该线程设置为当前线程，此时处于就绪状态。在线程运行之后，从等待或者睡眠中回来之后，也会处于就绪状态。

第三是**运行状态**。线程调度程序将处于就绪状态的线程设置为当前线程，此时线程就进入了运行状态，开始运行run函数当中的代码。

第四是**阻塞状态**。线程正在运行的时候，被暂停，通常是为了等待某个时间的发生(比如说某项资源就绪)之后再继续运行。sleep,suspend，wait等方法都可以导致线程阻塞。

第五是**死亡状态**。如果一个线程的run方法执行结束或者调用stop方法后，该线程就会死亡。对于已经死亡的线程，无法再使用start方法令其进入就绪。

### 二、start () 方法

1、为什么需要start方法；它的作用是什么？

**start () 方法来启动线程，真正实现了多线程运行。**  
start方法的作用就是将线程由NEW状态，变为RUNABLE状态。当线程创建成功时，线程处于NEW（新建）状态，如果你不调用start()方法，那么线程永远处于NEW状态。调用start()后，才会变为RUNABLE状态，线程才可以运行。

2、调用start () 方法后，线程是不是马上执行？

线程不是马上执行的；准确来说，调用start()方法后，线程的状态是“READY（就绪）”状态，而不是“RUNNING（运行中）”状态（关于线程的状态详细。线程要等待CPU调度，不同的JVM有不同的调度算法，线程何时被调度是未知的。因此，start () 方法的被调用顺序不能决定线程

#### 热门故事

当了三年上门女婿备受屈辱，今天我终于做了一回男人

一个故事告诉你，古代后宫不受宠的女人有多惨

如果最后那个人不是你，那么是谁都可以

当女人出现这些行为，说明她想放弃婚姻了

#### 推荐阅读

java面试题 --- 线程  
阅读 235

【高并发】深入理解线程的执行顺序  
阅读 319

【Java面试】由守护线程引发的一系列思考  
阅读 80

Thread各种state  
阅读 196

iOS多线程之NSThread  
阅读 226



# 深入浅出线程Thread类的start()方法和run()方法

只能调用start () 方法一次，多次启动一个线程是非法的。特别是当线程已经结束执行后，不能再重新启动。

## 三、run()方法

1、run方法又是一个什么样的方法？run方法与start方法有什么关联？

### run () 方法当作普通方法的方式调用

run()其实是一个普通方法，只不过当线程调用了start()方法后，一旦线程被CPU调度，处于运行状态，那么线程才会去调用这个run () 方法；

2、run () 方法的执行是不是需要线程调用start () 方法

上面说了，run () 方法是一个普通的对象方法，因此，不需要线程调用start () 后才可以调用的。可以线程对象可以随时随地调用run方法。

### #Example1:

```
1 Thread t1 = new Thread(new MyTask(1));
2 Thread t2 = new Thread(new MyTask(2));
3     t1.run();
4     t2.run();
```

上面的输出结果是固定的：

```
count的值: 1
count的值: 2
```

再看另一个实例：

```
1 Thread t1 = new Thread(new MyTask());
2 Thread t2 = new Thread(new MyTask());
3     t1.start();
4     t2.start();
```

这个输出结果不是固定的，因为线程的运行没法预测。运行结果可能不一样。

### MyTask 类:

```
1 //实现Runnable接口
2 class MyTask implements Runnable{
3
4     int count;
5     public MyTask(int count) {
6         this.count=count;
7     }
8     @Override
9     public void run() {
10        System.out.println("count的值: "+count);
11    }
12 }
```

### 热门故事

当了三年上门女婿备受屈辱，今天我终于做了一回男人

一个故事告诉你，古代后宫不受宠的女人有多惨

如果最后那个人不是你，那么是谁都可以

当女人出现这些行为，说明她想放弃婚姻了

### 推荐阅读

java面试题 --- 线程  
阅读 235

【高并发】深入理解线程的执行顺序  
阅读 319

【Java面试】由守护线程引发的一系列思考  
阅读 80

Thread各种state  
阅读 196

iOS多线程之NSThread  
阅读 226



# 深入浅出线程Thread类的start()方法和run()方法



程序员欧阳

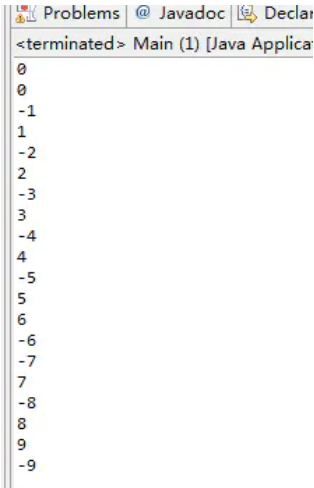
关注

赞赏支持

## 1、用start方法启动线程

```
1 public class Main {
2
3     public static void main(String[] args) {
4         Thread t1 = new Thread(new T1());
5         Thread t2 = new Thread(new T2());
6         t1.start();
7         t2.start();
8     }
9
10 }
11
12 class T1 implements Runnable {
13     public void run() {
14         try {
15             for(int i=0;i<10;i++){
16                 System.out.println(i);
17                 Thread.sleep(100); //模拟耗时任务
18             }
19         } catch (InterruptedException e) {
20             e.printStackTrace();
21         }
22     }
23 }
24
25 class T2 implements Runnable {
26     public void run() {
27         try {
28             for(int i=0;i>-10;i--){
29                 System.out.println(i);
30                 Thread.sleep(100); //模拟耗时任务
31             }
32         } catch (InterruptedException e) {
33             e.printStackTrace();
34         }
35     }
36 }
```

结果:



这里写图片描述

说明两线程是并发执行的。

## 2、先用run方法启动线程

将上面的start()改为run()

### 热门故事

当了三年上门女婿备受屈辱，今天我终于做了一回男人

一个故事告诉你，古代后宫不受宠的女人有多惨

如果最后那个人不是你，那么是谁都可以

当女人出现这些行为，说明她想放弃婚姻了

### 推荐阅读

java面试题 --- 线程  
阅读 235

【高并发】深入理解线程的执行顺序  
阅读 319

【Java面试】由守护线程引发的一系列思考  
阅读 80

Thread各种state  
阅读 196

iOS多线程之NSThread  
阅读 226



crm管理系统



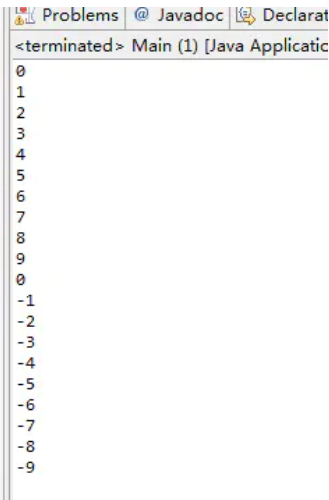
# 深入浅出线程Thread类的start()方法和run()方法

程序员欧阳

关注

赞赏支持

```
6         t1.run();
7         t2.run();
8     }
9
10 }
```



这里写图片描述

说明两线程实际是顺序执行的。

## 总结：

通过实例1和实例和我们可知start方法是用于启动线程的，可以实现并发，而run方法只是一个普通方法，是不能实现并发的，只是在并发执行的时候会调用。

说到这里，不知道小伙伴们有没有明白这两个方法的区别，如果还有疑问，可以留言交流。

## 四、start()方法和run()方法源码解析（基于JDK1.7.0\_40）

```
1 public synchronized void start() {
2     // 如果线程不是"就绪状态", 则抛出异常!
3     if (threadStatus != 0)
4         throw new IllegalArgumentException();
5     // 将线程添加到ThreadGroup中
6     group.add(this);
7     boolean started = false;
8     try {
9         // 通过start0()启动线程, 新线程会调用run()方法
10        start0();
11        // 设置started标记=true
12        started = true;
13    } finally {
14        try {
15            if (!started) {
16                group.threadStartFailed(this);
17            }
18        } catch (Throwable ignore) {}
19    }
20 }
21 }
```

```
1 public void run() {
2     if (target != null) {
3         target.run();
4     }
5 }
```

### 热门故事

当了三年上门女婿备受屈辱，今天我终于做了一回男人

一个故事告诉你，古代后宫不受宠的女人有多惨

如果最后那个人不是你，那么是谁都可以

当女人出现这些行为，说明她想放弃婚姻了

### 推荐阅读

java面试题 --- 线程  
阅读 235

【高并发】深入理解线程的执行顺序  
阅读 319

【Java面试】由守护线程引发的一系列思考  
阅读 80

Thread各种state  
阅读 196

iOS多线程之NSThread  
阅读 226



crm管理系统

# 深入浅出线程Thread类的start()方法和run()方法

## 五、真正理解Thread类

Thread类的对象其实也是一个java对象，只不过每一个Thread类的对象对应着一个线程。  
Thread类的对象就是提供给用户用于操作线程、获取线程的信息。真正的底层线程用户是看不到的了。  
因此，当一个线程结束了，死掉了，对应的Thread的对象仍能调用，除了start()方法外的所有方法（死亡的线程不能再次启动），如run()、getName()、getPriority () 等等

```
1 //简单起见，使用匿名内部类的方法来创建线程
2 Thread thread = new Thread(){
3     @Override
4     public void run() {
5         System.out.println("Thread对象的run方法被执行了");
6     }
7 };
8 //线程启动
9 thread.start();
10
11 //用循环去监听线程thread是否还活着，只有当线程thread已经结束了，才跳出循环
12 while(thread.isAlive()){
13     //线程thread结束了，但仍能调用thread对象的大部分方法
14     System.out.println("线程"+thread.getName()+"的状态: "+thread.getState()+"---优先级: "+thread
15 //调用run方法
16 thread.run();
17 //当线程结束时，start方法不能调用，下面的方法将会抛出异常
18 thread.start();
```

### 参考资料

- <http://www.cnblogs.com/jinggod/p/8485143.html>
- <https://blog.csdn.net/u010568463/article/details/47911181>
- <https://blog.csdn.net/xuxurui007/article/details/7685076>

文章有不当之处，欢迎指正，你也可以关注我的微信公众号：好好学java，获取优质学习资源。



3人点赞 >





随笔



热门故事

当了三年上门女婿备受屈辱，今天我终于做了一回男人

一个故事告诉你，古代后宫不受宠的女人有多惨

如果最后那个人不是你，那么是谁都可以

当女人出现这些行为，说明她想放弃婚姻了

推荐阅读

java面试题 --- 线程  
阅读 235

【高并发】深入理解线程的执行顺序  
阅读 319

【Java面试】由守护线程引发的一系列思考  
阅读 80

Thread各种state  
阅读 196

iOS多线程之NSThread  
阅读 226




更多精彩内容，就在简书APP



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



程序员欧阳 个人博客: <a href="https://links.jianshu.com/go?to=h...>https://links.jianshu.com/go?to=h...  
总资产83 共写了26.1W字 获得1,372个赞 共1,721个粉丝

关注

