

如何创建并运行java线程

[原文链接](#) 译者：章筱虎 校对：方腾飞

Java线程类也是一个object类,它的实例都继承自java.lang.Thread或其子类。可以用如下方式用java中创建一个线程：

```
1 | Tread thread = new Thread();
```

执行该线程可以调用该线程的start()方法:

```
1 | thread.start();
```

在上面的例子中，我们并没有为线程编写运行代码，因此调用该方法后线程就终止了。

编写线程运行时执行的代码有两种方式：一种是创建Thread子类的一个实例并重写run方法，第二种是创建类的时候实现Runnable接口。接下来我们会具体讲解这两种方法：

创建Thread的子类

创建Thread子类的一个实例并重写run方法，run方法会在调用start()方法之后被执行。例子如下：

```
1 | public class MyThread extends Thread {  
2 |     public void run(){  
3 |         System.out.println("MyThread running");  
4 |     }  
5 | }
```

可以用如下方式创建并运行上述Thread子类

```
1 | MyThread myThread = new MyThread();  
2 | myTread.start();
```

一旦线程启动后start方法就会立即返回，而不会等待到run方法执行完毕才返回。就好像run方法是在另外一个cpu上执行一样。当run方法执行后，将会打印出字符串MyThread running。

你也可以如下创建一个Thread的匿名子类：

```
1 | Thread thread = new Thread(){  
2 |     public void run(){  
3 |         System.out.println("Thread Running");  
4 |     }  
5 | };  
6 | thread.start();
```

当新的线程的run方法执行以后，计算机将会打印出字符串“Thread Running”。

实现Runnable接口

第二种编写线程执行代码的方式是新建一个实现了java.lang.Runnable接口的类的实例，实例中的方法可以被线程调用。下面给出例子：

```
1 public class MyRunnable implements Runnable {
2     public void run(){
3         System.out.println("MyRunnable running");
4     }
5 }
```

为了使线程能够执行run()方法，需要在Thread类的构造函数中传入 MyRunnable的实例对象。示例如下：

```
1 Thread thread = new Thread(new MyRunnable());
2 thread.start();
```

当线程运行时，它将会调用实现了Runnable接口的run方法。上例中将会打印出“MyRunnable running”。

同样，也可以创建一个实现了Runnable接口的匿名类，如下所示：

```
1 Runnable myRunnable = new Runnable(){
2     public void run(){
3         System.out.println("Runnable running");
4     }
5 }
6 Thread thread = new Thread(myRunnable);
7 thread.start();
```

创建子类还是实现Runnable接口？

对于这两种方式哪种好并没有一个确定的答案，它们都能满足要求。就我个人意见，我更倾向于实现Runnable接口这种方法。因为线程池可以有效地管理实现了Runnable接口的线程，如果线程池满了，新的线程就会排队等候执行，直到线程池空闲出来为止。而如果线程是通过实现Thread子类实现的，这将会复杂一些。

有时我们要同时融合实现Runnable接口和Thread子类两种方式。例如，实现了Thread子类的实例可以执行多个实现了Runnable接口的线程。一个典型的应用就是线程池。

常见错误：调用run()方法而非start()方法

创建并运行一个线程所犯的常见错误是调用线程的run()方法而非start()方法，如下所示：

```
1 Thread newThread = new Thread(MyRunnable());
2 newThread.run(); //should be start();
```

起初你并不会感觉到有什么不妥，因为run()方法的确如你所愿的被调用了。但是，事实上,run()方法并非是由刚创建的新线程所执行的，而是被创建新线程的当前线程所执行了。也就是被执行上面两行代码的线程所执行的。想要让创建的新线程执行run()方法，必须调用新线程的start方法。

线程名

当创建一个线程的时候，可以给线程起一个名字。它有助于我们区分不同的线程。例如：如果有多个线程写入System.out，我们就能够通过线程名容易的找出是哪个线程正在输出。例子如下：

```
1 MyRunnable runnable = new MyRunnable();
2 Thread thread = new Thread(runnable, "New Thread");
3 thread.start();
4 System.out.println(thread.getName());
```

需要注意的是，因为MyRunnable并非Thread的子类，所以MyRunnable类并没有getName()方法。可以通过以下方式得到当前线程的引用：

```
1 Thread.currentThread();
```

因此，通过如下代码可以得到当前线程的名字：

```
1 String threadName = Thread.currentThread().getName();
```

线程代码举例：

这里是一个小小的例子。首先输出执行main()方法线程名字。这个线程JVM分配的。然后开启10个线程，命名为1~10。每个线程输出自己的名字后就退出。

```
01 public class ThreadExample {
02     public static void main(String[] args){
03         System.out.println(Thread.currentThread().getName());
04         for(int i=0; i<10; i++){
05             new Thread("" + i){
06                 public void run(){
07                     System.out.println("Thread: " + getName() + "running");
08                 }
09             }.start();
10         }
11     }
12 }
```

需要注意的是，尽管启动线程的顺序是有序的，但是执行的顺序并非是有序的。也就是说，1号线程并不一定是第一个将自己名字输出到控制台的线程。这是因为线程是并行执行而非顺序的。Jvm和操作系统一起决定了线程的执行顺序，他和线程的启动顺序并非一定是一致的。

2013/03/24 10:45下午

通俗易懂！

Ulric Qin

2013/05/18 9:32下午

因为线程池可以有效的管理实现了Runnable接口的线程，如果线程池满了，新的线程就会排队等候执行，直到线程池空闲出来为止。而如果线程是通过实现Thread子类实现的，这将会复杂一些

初学，这句话没看懂，不知道为什么，求解释

java

2013/07/12 11:48上午

同求 不明白为什么继承Thread的话会更复杂

方 腾飞

2013/12/04 11:03上午

因为在java里，每个类只能有一个继承类，如果继承了Thread类就不能继承其他类了。

方 腾飞

2013/12/04 11:13上午

哪一句不明白？

E网情深

2013/10/22 4:41下午

因为线程池可以有效的管理实现了Runnable接口的线程，如果线程池满了，新的线程就会排队等候执行，直到线程池空闲出来为止。而如果线程是通过实现Thread子类实现的，这将会复杂一些

实现了Runnable接口，不是线程，应该称为任务，线程池只接受任务，线程池怎么还可以接受线程呢？求解

方 腾飞

2013/12/04 11:13上午

嗯 这里用任务更合适。

zhengJackson

2013/10/31 10:58下午

老大，文章中有段话：

但是，事实上，run()方法并非是由刚创建的新线程所执行的，而是被创建新线程的当前线程所执行了。也就是被执行上面两行代码的线程所执行的。想要让创建的新线程执行run()方法，必须调用新线程的start方法。

虽然很多人都说线程的调用是调用start()，不要直接用新创建的线程对象去调用线程的run()方法，我也只记得了这个结论。

文章那段话，我理解的并不是那么深刻，还是有点不太理解，可否再详细解锁一下？thanks

方 腾飞

2013/12/04 11:12上午

就是线程的实现决定的。启动线程使用start方法，执行代码写在run方法里。因为需要调用本地方法启动线程，

```
private native void start0();
```

匿名

2013/11/09 2:10下午

呵呵 小虎。

漂泊的小湖

2013/11/26 3:43下午

实现了Thread子类的实例可以执行多个实现了Runnable接口的线程。一个典型的应用就是线程池。

这个地方不明白 老大求解哇

方 腾飞

2013/12/04 11:09上午

提交给线程池的类，必须实现Runnable接口，这些类会被线程池里的线程来执行。

Darker

2018/11/13 3:57下午

你好！有没有简单例子，参考一下。谢谢

匿名

2014/04/09 4:56下午

//

方 腾飞：

提交给线程池的类，必须实现Runnable接口，这些类会被线程池里的线程来执行。

//

atshaonian

2018/03/01 9:28上午

通俗易懂，非常适合初学者