

java 创建线程的三种方法 Callable,Runnable, Thread 比较及用法

作者: [huihui](#)

原文链接: <https://ld246.com/article/1484558264415>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

编写多线程程序是为了实现多任务的并发执行，从而能够更好地与用户交互。一般有三种方法，Thread, Runnable, Callable.

Runnable和Callable的区别是，

(1)Callable规定的方法是call(),Runnable规定的方法是run()。

(2)Callable的任务执行后可返回值，而Runnable的任务是不能返回值得

(3)call方法可以抛出异常，run方法不可以

(4)运行Callable任务可以拿到一个Future对象，表示异步计算的结果。它提供了检查计算是否完成的方法，以等待计算的完成，并检索计算的结果。通过Future对象可以了解任务执行情况，可取消任务的行，还可获取执行结果。

1、通过实现Runnable接口来创建Thread线程：

步骤1：创建实现Runnable接口的类：

```
class SomeRunnable implements Runnable
{

    public void run()
    {
        //do something here
    }
}
```

步骤2：创建一个类对象：

```
Runnable oneRunnable = new SomeRunnable();
```

步骤3：由Runnable创建一个Thread对象：

```
Thread oneThread = new Thread(oneRunnable);
```

步骤4：启动线程：

```
oneThread.start();
```

至此，一个线程就创建完成了。

注释：线程的执行流程很简单，当执行代码oneThread.start();时，就会执行oneRunnable对象中的void run();方法，

该方法执行完成后，线程就消亡了。

2、与方法1类似，通过实现Callable接口来创建Thread线程：其中，Callable接口（也只有一个方法定义如下：

```
public interface Callable  
{  
  
    V call() throws Exception;  
  
}
```

步骤1：创建实现Callable接口的类SomeCallable（略）；

步骤2：创建一个类对象：

```
Callable oneCallable = new SomeCallable();
```

步骤3：由Callable创建一个FutureTask对象：

```
FutureTask oneTask = new FutureTask(oneCallable);
```

注释：FutureTask是一个包装器，它通过接受Callable来创建，它同时实现了Future和Runnable接口。

步骤4：由FutureTask创建一个Thread对象：

```
Thread oneThread = new Thread(oneTask);
```

步骤5：启动线程：

```
oneThread.start();
```

至此，一个线程就创建完成了。

3、通过继承Thread类来创建一个线程：

步骤1：定义一个继承Thread类的子类：

```
class SomeThead extends Thraad  
{  
  
    public void run()  
    {  
  
        //do something here  
  
    }  
}
```

```
}  
  
}
```

步骤2：构造子类的一个对象：

```
SomeThread oneThread = new SomeThread();
```

步骤3：启动线程：

```
oneThread.start();
```

至此，一个线程就创建完成了。