



java创建线程的三种方式及其对比

2018年1月14日 22:14:15

标签: java / 多线程 / 编程

91070



36

Java中创建线程主要有三种方式：

一、通过Thread类创建线程类

(1) Thread类的子类，并重写该类的run方法，该run方法的方法体就代表了线程要完成的任务。因此把run()方法称为执行体。

(2) 创建Thread子类的实例，即创建了线程对象。

(3) 调用线程对象的start()方法来启动该线程。

[java]

```
1. package com.thread;
2.
3. public class FirstThreadTest extends Thread{
4.     int i = 0;
5.     //重写run方法，run方法的方法体就是现场执行体
6.     public void run()
7.     {
8.         for(;i<100;i++){
9.             System.out.println(getName()+" "+i);
10.        }
11.    }
12. }
13. public static void main(String[] args)
14. {
15.     for(int i = 0;i< 100;i++)
16.     {
17.         System.out.println(Thread.currentThread().getName()+" "+i);
18.         if(i==20)
19.         {
20.             new FirstThreadTest().start();
21.             new FirstThreadTest().start();
22.         }
23.     }
24. }
25.
26. }
```

上述代码中Thread.currentThread()方法返回当前正在执行的线程对象。GetName()方法返回调用该方法的线程的名字。

二、通过Runnable接口创建线程类

(1) 定义Runnable接口的实现类，并重写该接口的run()方法，该run()方法的方法体同样是该线程的线程执行体。

(2) 创建 Runnable实现类的实例，并依此实例作为Thread的target来创建Thread对象，该Thread对象才是真正的线程对象。

(3) 调用线程对象的start()方法来启动该线程。

示例代码如下：

[java]

```
1. package com.thread;
2.
3. public class RunnableThreadTest implements Runnable
4. {
```

联系我们



请扫描二维码联系

webmaster@csdn.net

400-660-0108

QQ客服 在线客服

关于 招聘 广告服务 网站地图

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

广告

他的最新文章

javascript动态插入html元素

javascript同步执行逻辑

jquery中html、text、val的区别

地图标注随地图旋转而旋转

numpy函数学习

文章分类

C/C++

高性能计算

项目管理

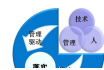
linux

数据结构与算法

机器学习

展开

博主专栏



技术人员谈管理

70250

24 篇



C++ 编程规范

82887

48 篇



Geoserver&O

系统

219926

文章存档

内容举报

2017年1月

2016年12月

2016年11月

2016年5月

返回顶部

```
8.     {
9.         for(i = 0;i <100;i++)
10.        {
11.            System.out.println(Thread.currentThread().getName()+" "+i);
12.        }
13.    }
14.    public static void main(String[] args)
15.    {
16.        for(int i = 0;i < 100;i++)
17.        {
18.            System.out.println(Thread.currentThread().getName()+" "+i);
19.            if(i==20)
20.            {
21.                RunnableThreadTest rtt = new RunnableThreadTest();
22.                new Thread(rtt,"新线程1").start();
23.                new Thread(rtt,"新线程2").start();
24.            }
25.        }
26.    }
27.
28.
29. }
```

三、通过Callable和Future创建线程

- (1) 创建Callable接口的实现类，并实现call()方法，该call()方法将作为线程执行体，并且有返回值。
- (2) 创建Callable实现类的实例，使用FutureTask类来包装Callable对象，该FutureTask对象封装了该Callable对象的call()方法的返回值。
- (3) 使用FutureTask对象作为Thread对象的target创建并启动新线程。
- (4) 调用FutureTask对象的get()方法来获得子线程执行结束后的返回值

实例代码：

```
[java]
1. package com.thread;
2.
3. import java.util.concurrent.Callable;
4. import java.util.concurrent.ExecutionException;
5. import java.util.concurrent.FutureTask;
6.
7. public class CallableThreadTest implements Callable<Integer>
8. {
9.
10.     public static void main(String[] args)
11.     {
12.         CallableThreadTest ctt = new CallableThreadTest();
13.         FutureTask<Integer> ft = new FutureTask<>(ctt);
14.         for(int i = 0;i < 100;i++)
15.         {
16.             System.out.println(Thread.currentThread().getName()+" 的循环变量i的值"+i);
17.             if(i==20)
18.             {
19.                 new Thread(ft,"有返回值的线程").start();
20.             }
21.         }
22.         try
23.         {
24.             System.out.println("子线程的返回值: "+ft.get());
25.         } catch (InterruptedException e)
26.         {
27.             e.printStackTrace();
28.         } catch (ExecutionException e)
29.         {
30.             e.printStackTrace();
31.         }
32.     }
33.
34.
35.     @Override
36.     public Integer call() throws Exception
37.     {
38.         int i = 0;
39.         for(;i<100;i++)
40.         {
41.             System.out.println(Thread.currentThread().getName()+" "+i);
42.         }
43.     }
44. }
```

联系我们



请扫描二维码联系

✉webmaster@csdn.net

☎400-660-0108

👤QQ客服 🗣客

关于 招聘 广告服务 网站地图

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

📖 42983

STL之五：set/multiset用法详解

📖 38036

STL之三：deque用法详解

📖 32348

JAVA中ListIterator和Iterator详解

📖 30397

python网络爬虫抓取图片

📖 29885



内容举报



返回顶部

46. | }

二、创建线程的三种方式的对比

采用实现Runnable、Callable接口的方式创见多线程时，优势是：

线程类实现了Runnable接口或Callable接口，还可以继承其他类。

在这种方式下，多个线程可以共享同一个target对象，所以非常适合多个相同线程来处理同一份资源的情况，从而可以将CPU、代码和数据分开，形成清晰的模型，较好地体现了面向对象的思想。

劣势是：

编程稍微复杂，如果要访问当前线程，则必须使用Thread.currentThread()方法。

使用继承Thread类的方式创建多线程时优势是：

编写简单，如果需要访问当前线程，则无需使用Thread.currentThread()方法，直接使用this即可获得当前线程。

劣势是：

线程类已经继承了Thread类，所以不能再继承其他父类。

联系我们



请扫描二维码联系

✉ webmaster@csdn.net

☎ 400-660-0108




👤 QQ客服 🗣 客服

关于 招聘 广告服务 网站地图

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

👤 目前您尚未登录，请 [登录](#) 或 [注册](#) 后进行评论

-  **zhang__yong** 2018-02-11 15:36 回复 9楼
言简意赅
-  **yong0717** 2018-01-10 10:45 回复 8楼
写的不错
-  **drancy** 2017-10-29 14:28 回复 7楼
字-学习

查看 12 条热评

创建线程的三种方法

 **baidu_21578557** 2016年04月16日 21:34 6002

第一种方法：继承Thread类，重写run()方法，run()方法代表线程要执行的任务。 第二种方法：实现Runnable接口，重写run()方法，run()方法代表线程要执行的任务。 第三种方法：实...

java线程（1）——三种创建线程的方式

 **u010066934** 2016年04月04日 21:16 1582

1、接口实现更灵活，java不支持多继承。在这方面，Runnable和Callable更有优势。 2、返回值问题。Runnable和Thread都不能有返回值，但Callable可以，而...

java创建线程的四种方式

 **u012973218** 2016年04月29日 10:51 3067

java创建线程的三种方式 1. 继承Thread类创建线程类 2. 通过Runnable接口创建线程类 3. 通过Callable和FutureTask创建线程 a. 创建Callab...

⚠ 内容举报

⬆ 返回顶部

Java创建线程的三种方式及其对比

 shenggaofei

2016年09月23日 20:54

🔖 3999

Java中创建线程主要有三种方式：一、继承Thread类创建线程类
(1) 定义Thread类的子类，并重写该类的run方法，该run方法的方法体就代表了线程要完成的任务。因此把run()...

创建线程的第三种方法-callable futureTask

 liguoqing2012

2015年09月23日 22:24

🔖 176

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;



创建线程的三种方法

 WUYUAN2011WOAINI

2012年05月11日 20:59

🔖 6460

MFC中有两类线程，分别称之为工作者线程和用户界面线程。二者的主要区别在于工作者线程没有消息循环，而用户界面线程有自己的消息队列和消息循环。工作者线程没有消息机制，通常用来执行后台计算和维护任...

创建线程的三种方法

 lixiaobuaa

2018年01月06日 16:59

🔖 26

方法一：继承Thread类 方法二：实现Runnable接口 方法三：实现Callable接口前两种方法需要重写run方法，第三种方法需要重写call方法。因为Java不允许多继承，允许实现多个接...

创建线程的三种方式优缺点

 sinat_27933301

2017年04月10日 11:08

🔖 1010

Java使用Thread类代表线程，所有的线程对象都必须是Thread类或其子类的实例。一、继承Thread类创建线程类 1.重写run方法。该run（）方法的方法体就代表了线程需要完成的任务 ...

Java创建线程的三种方式及其对比

 hellorichen

2016年11月11日 13:18

🔖 3644

Java中创建线程主要有三种方式：一、继承Thread类创建线程类 二、通过Runnable接口创建线程类 三、通过Callable和FutureTask创建线程 参考：http://blog....

Java创建线程的三种方式

 sinat_28028941

2017年03月07日 21:14

🔖 972

Java有三种创建线程的方式，分别是继承Thread类、实现Runnable接口和使用线程池 1、继承Thread类 使用该方式创建及使用线程需按以下三个步骤：（1）定义Thread类的子类，并...

创建线程的几种方法及比较

 u012470138

2016年11月21日 10:17

🔖 2614

1、通过继承Thread类创建线程 (1).首先定义一个类去继承Thread父类，重写父类中的run()方法。在run()方法中加入具体的任务代码或处理逻辑。(2).直接创建一个ThreadTes...

多线程编程——创建线程的三种方法

 quanzheng92

2014年11月08日 22:07

🔖 646

/** * 获取网络数据、处理大批量数据、使用到一些比较耗时的算法时，需要使用子线程处理 */ 1、创建子线程有三种方法（1）、NSThread（2）、NSOperat...

创建线程有几种不同的方式？

 wang_xing1993

2017年04月20日 10:45

🔖 2756

①继承Thread类（真正意义上的线程类），是Runnable接口的实现。 ②实现Runnable接口，并重写里面的run方法。 ③使用Executor框架创建线程池。Executor框架是juc...

创建线程的两种方式区别

 Krito_blog

2017年02月01日 12:49

🔖 1653

联系我们



请扫描二维码联系

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗣 客服中心

关于 招聘 广告服务 网站地图

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



内容举报



返回顶部

HashTable和HashMap的区别详解

fujiakai

2016年06月04日 19:35

41062

一、HashMap简介 HashMap是基于哈希表实现的，每一个元素是一个key-value对，其内部通过单链表解决冲突问题，容量不足（超过了阈值）时，同样会自动增长。 Has...

创建和 线程的两种方式

caidie_huang

2016年10月07日 10:52

10828

方式1：继承Thread类 步骤: 1):定义一个类A继承于java.lang.Thread类. 2):在A类中覆盖Thread类中的run方法. 3):我们在run方法中编写要执行的操作：...

创建线程的三种方式

scgaliguodong123_

2015年05月27日 17:13

6425

一、继承Thread类+重写run()方法 启动：创建子类对象+对象.start() 二、实现Runnable接口+重写run方法 启动：使用静态代理 1)、创建真实角色 2)、创建代理角色...

MFC中如何创建一个线程

liudonghong128

2016年12月29日 14:01

1038

MFC中如何创建一个线程.hCWinThread* FHZ_Thread;//声明线程 static UINT StartTestGKThread(void *param);//声明线程函数.c...

JAVA多线程实现的三种方式

aboy123

2014年07月31日 18:34

452407

JAVA多线程实现方式主要有三种：继承Thread类、实现Runnable接口、使用ExecutorService、Callable、Future实现有返回结果的多线程。其中前两种方式线程执行完后都没...

开启子线程的三种方式

u014756517

2016年07月20日 14:04

4836

package com.you;import java.util.Date;/** * 多线程的实现方式 * 1、继承Thread类 * 2、实现Runnable接口 * @autho...

联系我们



请扫描二维码联系

 webmaster@c

 400-660-0108

 QQ客服  客

关于 招聘 广告服务

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心