链滴

【并发编程】Java 如何开启一个线程（Thread,Runnable,Callable）

作者：moonce
原文链接：https://ld246.com/article/1541571725085
来源网站：链滴

Java如何开启一个线程

作者:Moonce

博客:望舒阁

一、三种方式

- 继承Thread
- 实现Runnable接口
- 实现Callable接口

二、实现Demo

```
package com.moonce.thread;

import java.util.concurrent.Callable;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.FutureTask;

/**
 *@author moonce
 *
 *类说明：新建Java线程
 */
public class ThreadClient  {
    /*扩展自Thread类*/
    private static class UseThread extends Thread {
        @Override
        public void run() {
            System.out.println("I am extends Thread");
        }
    }
    /*实现Runnable接口*/
    private static class UseRun implements Runnable{

        @Override
        public void run() {
            System.out.println("I am implements Runnable");
        }

    }

    /*实现Callable接口，允许有返回值*/
    private static class UseCall implements Callable<String>{

        @Override
        public String call() throws Exception {
            System.out.println("I am implements Callable");
            return "Callable Result";
        }
```

```
    }

    public static void main(String[] args)
            throws InterruptedException, ExecutionException {
        //开启UseThread线程
        UseThread useThread = new UseThread();
        useThread.start();

        //开启UseRun线程
        UseRun useRun = new UseRun();
        new Thread(useRun).start();

        //开启UseCall线程
        UseCall useCall = new UseCall();
        FutureTask<String> futureTask = new FutureTask<>(useCall);
        new Thread(futureTask).start();
        System.out.println(futureTask.get());//打印UseCall线程返回结果
    }
}
```

三、使用推荐

由于Java只支持单继承，所以用继承的方式创建线程，不够灵活，所以不推荐使用继承Thread；用实接口的方式创建线程，可以实现多个接口，下面是Runnable和Callable接口的区别，具体选用哪种需参照实际开发情况。

1. Callable重写的方法是call(),Runnable重写的方法是run();

2. Callable的任务执行后可返回值，而Runnable不能返回值；

3. call方法可以抛出异常，run()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，它供检查计算是否完成的方法以等待计算完成，并检索计算的结果。通过Future对象可以了解任务的执行情况，可取消任务的执行还可以获取执行的结果。