



链滴

1、创建线程的四种方式

作者: [hantiaotiao](#)

原文链接: <https://ld246.com/article/1677676320497>

来源网站: [链滴](#)

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

创建线程的四种方式

本质上都是实现Runnable接口

1、继承Thread类

```
public class TestThread extends Thread{
    public static void main(String[] args) {
        TestThread tt = new TestThread();
        tt.start();
    }

    @Override
    public void run() {
        super.run();
        System.out.println("New Thread");
    }
}
```

缺点：继承Thread后，无法继承其它类

2、实现Runnable接口

```
public class TestThread2 implements Runnable{
    public static void main(String[] args) {
        Thread thread = new Thread(new TestThread2());
        thread.start();

        new Thread(new Runnable(){
            @Override
            public void run() {
                System.out.println("匿名内部类");
            }
        }).start();

        //函数式接口->一个接口中只有一个方法，可以转换为lambda表达式 (java8新特性)
        new Thread(()-> System.out.println("lambda表达式")).start();
    }

    @Override
    public void run() {
        System.out.println("new thread");
    }
}
```

接口可以多继承

3、实现Callable接口

```
public class TestThread3 implements Callable<String> {

    public static void main(String[] args) throws ExecutionException, InterruptedException {
        FutureTask<String> ft = new FutureTask<>(new TestThread3());
        Thread thread = new Thread(ft);
        thread.start();
        //阻塞式的获取线程执行结果
        System.out.println(ft.get());
    }

    public String call(){
        return "你好";
    }
}
```

4、线程池

```
public class TestThread4 implements Runnable{

    public static void main(String[] args) {
        ExecutorService executorService = Executors.newFixedThreadPool(10);
        executorService.execute(new TestThread4());
    }

    @Override
    public void run() {
        System.out.println("你好 世界");
    }
}
```

不建议使用Executors创建线程池

底层是无界阻塞队列有可能造成OOM之外，我们使用Executors来创建线程池也不能自定义线程的名，不利于排查问题，所以建议直接使用ThreadPoolExecutor来定义线程池，这样可以灵活控制。****