

表11-1线程的常用方法

方法	功能
isAlive()	判断线程是否还“活”着，即线程是否还未终止
getPriority()	获得线程的优先级数值
setPriority()	设置线程的优先级数值
setName()	给线程一个名字
getName()	取得线程的名字
currentThread()	取得当前正在运行的线程对象，也就是取得自己本身

线程的常用方法一

```

1 public class TestThread {
2     public static void main(String[] args) throws Exception {
3         Runnable r = new MyThread();
4         Thread t = new Thread(r, "Name test");//定义线程对象，并传入参
5         数；
6         t.start();//启动线程；
7         System.out.println("name is: " + t.getName());//输出线程名称；
8         Thread.currentThread().sleep(5000);//线程暂停5分钟；
9         System.out.println(t.isAlive());//判断线程还在运行吗？
10        System.out.println("over!");
11    }
12 }
13 class MyThread implements Runnable {
14     //线程体；
15     public void run() {
16         for (int i = 0; i < 10; i++)
17             System.out.println(i);
18     }
19 }

```

线程的优先级

1. 处于就绪状态的线程，会进入“就绪队列”等待JVM来挑选。
2. 线程的优先级用数字表示，范围从1到10，一个线程的缺省优先级是5。
3. 使用下列方法获得或设置线程对象的优先级。

```
int getPriority();
```

```
void setPriority(int newPriority);
```

注意：优先级低只是意味着获得调度的概率低。并不是绝对先调用优先级高的线程后调用优先级低的线程。

线程的常用方法二

```

1 public class TestThread {
2     public static void main(String[] args) {
3         Thread t1 = new Thread(new MyThread(), "t1");
4         Thread t2 = new Thread(new MyThread(), "t2");

```

```
5         t1.setPriority(1);
6         t2.setPriority(10);
7         t1.start();
8         t2.start();
9     }
10 }
11 class MyThread extends Thread {
12     public void run() {
13         for (int i = 0; i < 10; i++) {
14             System.out.println(Thread.currentThread().getName() + ": " +
15                 i);
16         }
17     }
18 }
```