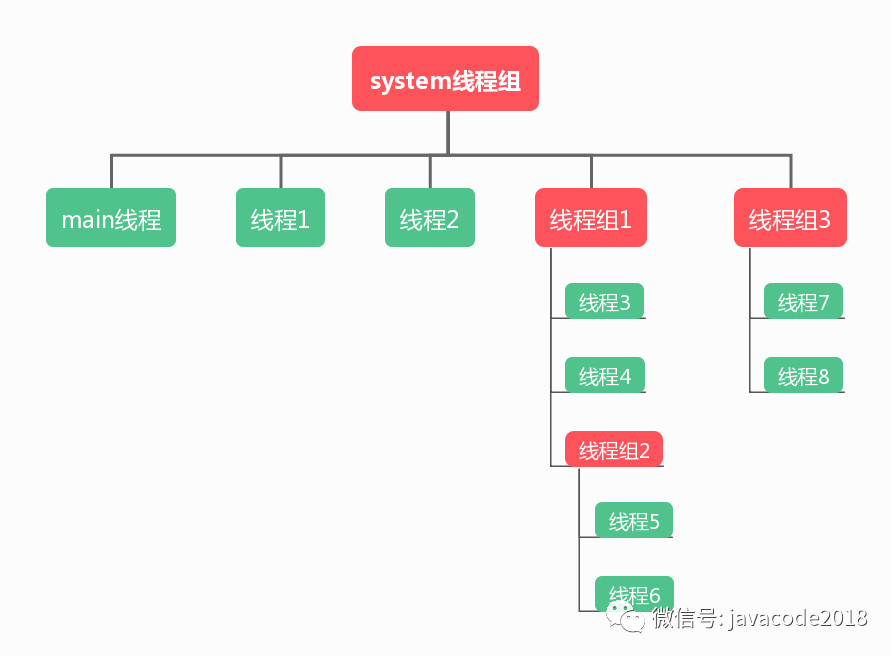
# 线程组

## 线程组

我们可以把线程归属到某个线程组中，线程组可以包含多个线程以及线程组，线程和线程组组成了父子关系，是个树形结构，如下图：



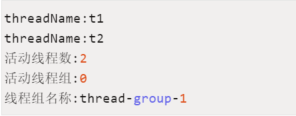
使用线程组可以方便管理线程，线程组提供了一些方法方便方便我们管理线程。

## 创建线程关联线程组

创建线程的时候，可以给线程指定一个线程组，代码如下：



输出结果：



activeCount()方法可以返回线程组中的所有活动线程数，包含下面的所有子孙节点的线程，由于线程组中的线程是动态变化的，这个值只能是一个估算值。

## 为线程组指定父线程组

创建线程组的时候，可以给其指定一个父线程组，也可以不指定，如果不指定父线程组，则父线程组为当前线程的线程组，java api有2个常用的构造方法用来创建线程组：



第一个构造方法未指定父线程组，看一下内部的实现：

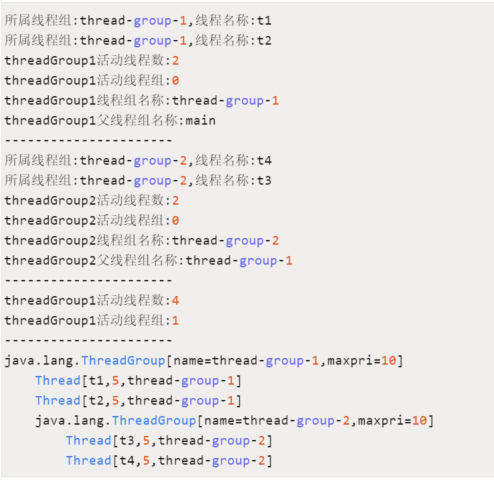


系统自动获取当前线程的线程组作为默认父线程组。

上一段示例代码：



输出结果：



代码解释：

1、threadGroup1未指定父线程组，系统获取了主线程的线程组作为threadGroup1的父线程组，输出结果中是：main

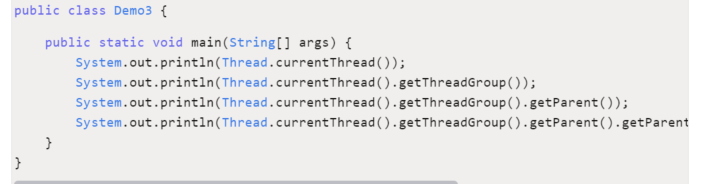
2、threadGroup1为threadGroup2的父线程组

3、threadGroup1活动线程数为4，包含了threadGroup1线程组中的t1、t2，以及子线程组threadGroup2中的t3、t4

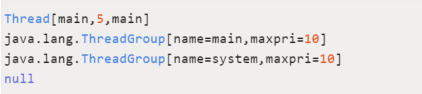
4、线程组的list()方法，将线程组中的所有子孙节点信息输出到控制台，用于调试使用

## 根线程组

获取根线程组：



运行上面代码，输出：

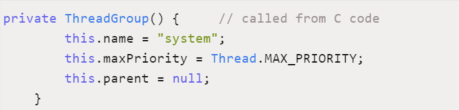


从上面代码可以看出：

1、主线程的线程组为main

2、根线程组为system

看一下ThreadGroup的源码：



发现ThreadGroup默认构造方法是private的，是由c调用的，创建的正是system线程组。

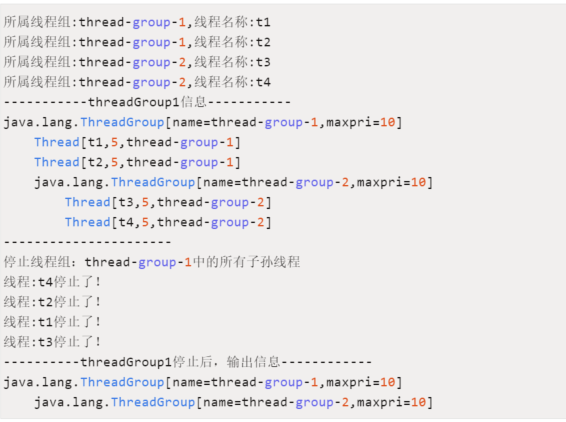
## 批量停止线程

调用线程组interrupt()，会将线程组树下的所有子孙线程中断标志置为true，可以用来批量中断线程。

示例代码：



输出：



停止线程之后，通过list()方法可以看出输出的信息中不包含已结束的线程了。

多说几句，建议大家再创建线程或者线程组的时候，给他们取一个有意义的名字，对于计算机来说，可能名字并不重要，但是在系统出问题的时候，你可能会去查看线程堆栈信息，如果你看到的都是t1、t2、t3，估计自己也比较崩溃，如果看到的是httpAccpHandler、dubboHandler类似的名字，应该会好很多。