

JAVA.LANG.THREADGROUP.INTERRUPT METHOD

http://www.tutorialspoint.com/java/lang/threadgroup_interrupt.htm

Copyright © tutorialspoint.com

Description

The **java.lang.ThreadGroup.interrupt** method interrupts all threads in this thread group.

Declaration

Following is the declaration for **java.lang.ThreadGroup.interrupt** method

```
public final void interrupt()
```

Parameters

- **NA**

Return Value

This method does not return any value.

Exception

- **SecurityException** -- if the current thread is not allowed to access this thread group or any of the threads in the thread group.

Example

The following example shows the usage of java.lang.ThreadGroup.interrupt method.

```
package com.tutorialspoint;

import java.lang.*;

class newThread extends Thread {
    boolean stop;

    newThread(ThreadGroup group, String name) {
        super(group, name);
        stop = false;
    }

    public void run() {
        System.out.println(Thread.currentThread().getName() + " starting.");
        try {
            for(int i = 1; i < 1000; i++) {
                Thread.sleep(500);
                synchronized(this) {
                    if(stop)
                        break;
                }
            }
        } catch(Exception e) {
            System.out.print(Thread.currentThread().getName());
            System.out.println(" interrupted.");
        }
        System.out.println(Thread.currentThread().getName() + " exiting.");
    }

    synchronized void stopFunc() {
        stop = true;
    }
}
```

```

public class ThreadGroupDemo {

    public static void main(String args[]) throws Exception {

        ThreadGroup group = new ThreadGroup("new Group");

        newThread t1 = new newThread(group, "Thread1");
        newThread t2 = new newThread(group, "Thread2");

        // this will call run() method
        t1.start();
        t2.start();

        Thread.sleep(1000);

        // it shows current active threads in Thread Group
        System.out.println(group.activeCount() + " threads in thread group...");

        // returns the number of thread groups
        Thread th[] = new Thread[group.activeCount()];
        group.enumerate(th);
        for(Thread t : th)
            System.out.println(t.getName());

        t1.stopFunc();
        Thread.sleep(1000);

        System.out.println(group.activeCount() + " threads in thread group...");
        // thread group interrupted
        group.interrupt();
    }
}

```

Let us compile and run the above program, this will produce the following result:

```

Thread1 starting.
Thread2 starting.
2 threads in thread group...
Thread1
Thread2
Thread1 exiting.
1 threads in thread group...
Thread2 interrupted.
Thread2 exiting.

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js