

JAVA.LANG.THREADGROUP.UNCAUGHTEXCEPTION METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.ThreadGroup.uncaughtException** method is called by the Java Virtual Machine when a thread in this thread group stops because of an uncaught exception, and the thread does not have a specific Thread.UncaughtExceptionHandler installed.

Declaration

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

```
public void uncaughtException(Thread t, Throwable e)
```

Parameters

- **t** -- This is the thread that is about to exit.
- **e** -- This is the uncaught exception.

Return Value

This method does not return any value.

Exception

- **NA**

Example

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

```
package com.tutorialspoint;

import java.lang.*;

public class ThreadGroupDemo implements Runnable {

    public static void main(String[] args) {
        ThreadGroupDemo tg = new ThreadGroupDemo();
        tg.func();
    }

    public void func()
    {
        try {
            // create a new ThreadGroup and a child for that ThreadGroup.
            newThreadGroup pGroup = new newThreadGroup("ParentThreadGroup");
            newThreadGroup cGroup = new newThreadGroup
                (pGroup, "ChildThreadGroup");

            // create another thread
            Thread thr2 = new Thread(pGroup, this);
            System.out.println("Starting " + thr2.getName() + "...");
            // this will call run() method
            thr2.start();

            // create third thread
            Thread thr3 = new Thread(cGroup, this);
            System.out.println("Starting " + thr3.getName() + "...");
            // this will call run() method
            thr3.start();
        }
    }
}
```

```

        try {
            Thread.sleep(500);
        }
        catch(InterruptedException ex) {}

        // interrupt the two threads
        thr2.interrupt();
        thr3.interrupt();

        // block until the other threads finish
        thr2.join();
        thr3.join();
    }
    catch(InterruptedException e) {
        System.out.println(e.toString());
    }
}

public void run() {
    try {
        System.out.print(Thread.currentThread().getName());
        System.out.println(" executing...");

        while(true) {
            Thread.sleep(500);
        }
    }
    catch(InterruptedException e) {
        Thread currThread = Thread.currentThread();
        System.out.print(currThread.getName());
        System.out.println(" interrupted:" + e.toString());

        // rethrow the exception
        throw new RuntimeException(e.getMessage());
    }
}
}

class newThreadGroup extends ThreadGroup {

    newThreadGroup(String n) {
        super(n);
    }

    newThreadGroup(ThreadGroup parent, String n) {
        super(parent, n);
    }

    public void uncaughtException(Thread t, Throwable e) {
        System.out.println(t + " has unhandled exception:" + e);
    }
}

```

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

```

Starting Thread-0...
Starting Thread-1...
Thread-0 executing...
Thread-1 executing...
Thread-0 interrupted: java.lang.InterruptedException: sleep interrupted
Thread[Thread-0,5,ParentThreadGroup] has unhandled exception: java.lang.RuntimeException:
sleep interrupted
Thread-1 interrupted: java.lang.InterruptedException: sleep interrupted
Thread[Thread-1,5,ChildThreadGroup] has unhandled exception: java.lang.RuntimeException:
sleep interrupted
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js

```