

Java中限时线程回调方式的实现 PDF转换可能丢失图片或格式
， 建议阅读原文

https://www.100test.com/kao_ti2020/462/2021_2022_Java_E4_B8_AD_E9_99_90_c104_462413.htm 线程回调方式我们已经在"使用回调和线程处理一个耗时响应过程"文中进行了讲述，但是有些情况下用户希望在指定时间内返回一个结果，免得无休止的等待下去。这时我们需要使用"限时线程回调方式"，它在原有线程回调的基础上加上了一个Timer以计算消耗的时间，如果时间期限到了任务还没有执行完的话即中断线程，示例代码如下：

```
package com.sitinspring.import
java.awt.event.ActionEvent.import
java.awt.event.ActionListener.import javax.swing.Timer./** **/** *
定时回调线程类 * * @author sitinspring(junglesong@gmail.com) *
* @date 2007-11-6 */public class TimedCallbackThread implements
Runnable { // 一秒的毫秒数常量 private final static int
ONE_SECOND = 1000. // 限制时间,以秒为单位 private final int
waitTime. // 已经流逝的时间 private int passedTime. private Timer
timer. private Thread thread. private MvcTcModel model. private
MvcTcView view. public TimedCallbackThread(MvcTcModel
model, MvcTcView view, int waitTime) { this.model = model.
this.view = view. this.waitTime = waitTime. this.passedTime = 0. //
创建并启动定时器 timer = new Timer(ONE_SECOND, new
ActionListener() { public void actionPerformed(ActionEvent evt) {
timeListener(). } }). timer.start(). // 创建并启动线程来完成任务
thread = new Thread(this). thread.start(). } private void
timeListener() { passedTime . // 动态显示状态 int modSeed =
```

```
passedTime % 3. if (modSeed == 0) { view.getLabel2().setText("响  
应中"). } else if (modSeed == 1) { view.getLabel2().setText("响应  
中.."). } else if (modSeed == 2) { view.getLabel2().setText("响应  
中."). } // 如果流逝时间大于规定时间则中断线程 if  
(passedTime > waitTime) { passedTime = waitTime.  
thread.interrupt(). } } public void run() { while (passedTime 执行效  
果如下： 100Test 下载频道开通，各类考试题目直接下载。详  
细请访问 www.100test.com
```