

Java线程超时控制的实现Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/594/2021\\_2022\\_Java\\_E7\\_BA\\_BF\\_E7\\_A8\\_8B\\_c104\\_594677.htm](https://www.100test.com/kao_ti2020/594/2021_2022_Java_E7_BA_BF_E7_A8_8B_c104_594677.htm) 一.何时需要超时控制 超时控制

一般使用阻塞时间比较长的操作上,有可能是和远程数据库的连接,也有可能是网络下载,在程序超时后,往往需要进行一些操作,比如退出线程,或者重新执行. 二.实现方法 方法1.(原创)

实现描述:使用一个守护线程作为计时器,并且在计时结束时抛出一个未检测异常。 具体实现: 类1:守护线程类 /\*\* \* 本线程设置了一个超时时间 \* 该线程开始运行后,经过指定超时时间, \* 该线程会抛出一个未检查异常通知调用该线程的程序超时 \* 在超时结束前可以调用该类的cancel方法取消计时 \*

```
@author solonote */ public class TimeoutThread extends Thread{  
/** * 计时器超时时间 */ private long timeout. /** * 计时是否被取消 */ private boolean isCanceled = false. /** * 当计时器超时时抛出的异常 */ private TimeoutException timeoutException. /** * 构造器 * @param timeout 指定超时的时间 */ public
```

```
TimeoutThread(long timeout, TimeoutException timeoutErr) {  
super(). this.timeout = timeout. this.timeoutException = timeoutErr.  
//设置本线程为守护线程 this.setDaemon(true). } /** * 取消计时 */ public synchronized void cancel() { isCanceled = true. } /** * 启动超时计时器 */ public void run() { try { Thread.sleep(timeout).  
if(!isCanceled) throw timeoutException. } catch
```

```
(InterruptedException e) { e.printStackTrace(). } } } 注: 类一中的
```

TimeoutException是下边的用户自定义类,不是java中的

java.util.concurrent.TimeoutException 类2.抛出异常类,该类继

承了RuntimeException，原因是run方法不能抛出已检测异常。

```
public class TimeoutException extends RuntimeException { /** * 序列化号 */ private static final long serialVersionUID = -8078853655388692688L. public TimeoutException(String errMessage) { super(errMessage). } } 使用方法: //初始化超时类 TimeoutThread t = new TimeoutThread(5000,new TimeoutException("超时")). try{ t.start(). .....要检测超时的程序段.... t.cancel(). }catch (TimeoutException e) { ...对超时的处理... }
```

TimeoutException可以更换为其他未检查异常类。方法说明: 本方法的使用可以实现线程自己管理超时,并且可以管理某一段代码超时,可以在方法内部给出处理办法。但是需要注意的是:本方法的超时时间并不是当前线程的运行时间,而是计时器开始计时起系统运行的时间。方法2: 是用join方法,在外部管理线程超时。Thread对象有一个join(long millis)方法,执行该方法时如果另一个线程中断了当前线程,就会抛出一个InterruptedException异常。可以采用这个机制在一个线程的外部处理这个线程的异常。我觉得这个方法很有局限性,因为是在外部定义该线程的方法,就只能对整个线程的run方法做出线程超时控制,而并不是对run方法中的某一段代码。并且该方法处理线程超时的处理方法也只能写在线程的外部。不过这个方法的超时定义是当前线程的运行时间。至于采用何种方法,根据个人的应用而定吧。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 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](http://www.100test.com)