

线程基础---wait(),notify的应用一例 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E7_BA_BF_E7_A8_8B_E5_9F_BA_E7_c104_145511.htm 本例子实现了两个线程，每个线程输出1到100的数字。第一个线程输出1-10，停止，通知第二个线程输出1-10 第二个线程停止 通知第一个线程输出11-20 ...实现的要点是在Java中，每个对象都有个对象锁标志(Object lock flag)与之想关联，当一个线程A调用对象的一段synchronized代码时，它首先要获取与这个对象关联的对象锁标志，然后执行相应的代码，执行结束后，把这个对象锁标志返回给对象；因此，在线程A执行synchronized代码期间，如果另一个线程B也要执行同一对象的一段synchronized代码时（不一定与线程A执行的相同），它将要等到线程A执行完后，才能继续...如何利用wait() notify() notifyAll()?

在synchronized代码被执行期间，线程可以调用对象的wait()方法，释放对象锁标志，进入等待状态，并且可以调用notify()或者notifyAll()方法通知正在等待的其他线程。notify()通知等待队列中的第一个线程，notifyAll()通知的是等待队列中的所有线程。

```
package jdeveloper.study./*  
*** Title: Jdevelopers Java  
Project* Description: n/a* Copyright: Copyright (c) 2001*
```

```
Company: soho http://www.ChinaJavaWorld.com* @author  
jdeveloper@21cn.com* @version 1.0*/import
```

```
java.lang.Runnable.import java.lang.Thread.public class
```

```
DemoThread implements Runnable{ public DemoThread() {
```

```
TestThread testthread1 = new TestThread(this,"1"). TestThread
```

```
testthread2 = new TestThread(this,"2"). testthread2.start().
```

```
testthread1.start(). } public static void main(String[] args) {
DemoThread demoThread1 = new DemoThread(). } public void
run(){ TestThread t = (TestThread) Thread.currentThread(). try{ if
(!t.getName().equalsIgnoreCase("1")) { synchronized(this) { wait().
} } while(true){ System.out.println("@time in thread" t.getName()
"=" t.increaseTime()). if(t.getTime() == 0) { synchronized(this) {
System.out.println("*****").
notify(). if ( t.getTime()==100 ) break. wait(). } } }catch(Exception
e){e.printStackTrace().} }}class TestThread extends Thread{ private
int time = 0 . public TestThread(Runnable r,String name){
super(r,name). } public int getTime(){ return time. } public int
increaseTime (){ return time. } 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
```