计算机二级辅导:Java线程新特征(条件变量)计算机等级考试
 条件变量是Java5线程中很重要的一个概念，顾名思义，条件变量就是表示条件的一种变量。但是必须说明，这里的条件是没有实际含义的，仅仅是个标记而已，并且条件的含义往往通过代码来赋予其含义。 这里的条件和普通意义上的条件表达式有着天壤之别。 条件变量都实现了java.util.concurrent.locks.Condition接口，条件变量的实例化是通过一个Lock对象上调用newCondition()方法来获取的，这样，条件就和一个锁对象绑定起来了。因此，Java中的条件变量只能和锁配合使用，来控制并发程序访问竞争资源的安全。 条件变量的出现是为了更精细控制线程等待与唤醒，在Java5之前，线程的等待与唤醒依靠的是Object对象的wait()和notify()/notifyAll()方法，这样的处理不够精细。 而在Java5中，一个锁可以有多个条件，每个条件上可以有多个线程等待，通过调用await()方法，可以让线程在该条件下等待。当调用signalAll()方法，又可以唤醒该条件下的等待的线程。有关Condition接口的API可以具体参考JavaAPI文档。 条件变量比较抽象，原因是他不是自然语言中的条件概念，而是程序控制的一种手段。 下面以一个银行存取款的模拟程序为例来揭盖Java多线程条件变量的神秘面纱：有一个账户，多个用户（线程）在同时操作这个账户，有的存款有的取款，存款随便存，取款有限制，不能透支，任何试图透支的操作都将等待里面有足够存款才执行操作。 import java.util.concurrent.ExecutorService. import

java.util.concurrent.Executors. import java.util.concurrent.locks.Condition. import java.util.concurrent.locks.Lock. import java.util.concurrent.locks.ReentrantLock. /** * Java线程：条件变量 * * @author leizhimin 2009-11-5 10:57:29 */ public class Test { public static void main(String[] args) { //创建并发访问的账户 MyCount myCount = new MyCount("9559920090121552 2", 10000). //创建一个线程池 ExecutorService pool = Executors.newFixedThreadPool(2). Thread t1 = new SaveThread("张三", myCount, 2000). Thread t2 = new SaveThread("李四", myCount, 3600). Thread t3 = new DrawThread("王五", myCount, 2700). Thread t4 = new SaveThread("老张", myCount, 600). Thread t5 = new DrawThread("老牛", myCount, 1300). Thread t6 = new DrawThread("胖子", myCount, 800). //执行各个线程 pool.execute(t1). pool.execute(t2). pool.execute(t3). pool.execute(t4). pool.execute(t5). pool.execute(t6). //关闭线程池 pool.shutdown(). } } /** * 存款线程类 */ class SaveThread extends Thread { private String name. //操作人 private MyCount myCount. //账户 private int x. //存款金额 SaveThread(String name, MyCount myCount, int x) { this.name = name. this.myCount = myCount. this.x = x. } public void run() { myCount.saving(x, name). } } /** * 取款线程类 */ class DrawThread extends Thread { private String name. //操作人 private MyCount myCount. //账户 private int x. //存款金额 DrawThread(String name, MyCount myCount, int x) { this.name = name. this.myCount = myCount. this.x = x. } public void run() { myCount.drawing(x, name). } } /** *

普通银行账户，不可透支 */ class MyCount { private String oid. //账号 private int cash. //账户余额 private Lock lock = new ReentrantLock(). //账户锁 private Condition _save = lock.newCondition(). //存款条件 private Condition _draw = lock.newCondition(). //取款条件 MyCount(String oid, int cash) { this.oid = oid. this.cash = cash. } /** * 存款 * * @param x 操作金额 * @param name 操作人 */ public void saving(int x, String name) { lock.lock(). //获取锁 if (x 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