Java线程:并发协作(死锁)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_ti2020/644/2021_2022_Java_E7_BA_BF_E7_A8_8B_c104_644666.htm 线程发生死锁可能性很小，即使看似可能发生死锁的代码，在运行时发生死锁的可能性也是小之又小。 发生死锁的原因一般是两个对象的锁相互等待造成的。 在《Java线程：线程的同步与锁》一文中，简述死锁的概念与简单例子，但是所给的例子是不完整的，这里给出一个完整的例子。 /** * Java线程：并发协作-死锁 * * @author Administrator 2009-11-4 22:06:13 */ public class Test { public static void main(String[] args) { DeadlockRisk dead = new DeadlockRisk(). MyThread t1 = new MyThread(dead, 1, 2). MyThread t2 = new MyThread(dead, 3, 4). MyThread t3 = new MyThread(dead, 5, 6). MyThread t4 = new MyThread(dead, 7, 8). t1.start(). t2.start(). t3.start(). t4.start(). } } class MyThread extends Thread { private DeadlockRisk dead. private int a, b. MyThread(DeadlockRisk dead, int a, int b) { this.dead = dead. this.a = a. this.b = b. } @Override public void run() { dead.read(). dead.write(a, b). } } class DeadlockRisk { private static class Resource { public int value. }