东华大学继续教育学院

2020年秋季学期远程学历教育《Java程序设计》期末大作业

班级 19秋计算机专业 姓名 曹会金 学号 619945272067 成绩

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 项目 | 一 | 二 | 三 | 四 | 五 | 六 | 总分 |
| 成绩 |  |  |  |  |  |  |  |

1. 选择题（本大题共10小题，每小题 1分， 共10分）
2. 下列哪个不是面向对象程序设计的基本特征（ C ）
3. 封装性
4. 继承性
5. 安全性
6. 多态性
7. 关于Java程序设计语言，不正确的说法是（ A ）
8. 垃圾回收困难
9. 健壮性强
10. 支持多线程编程
11. 安全性较高
12. 关于类，不正确的是（D）
13. 类是Java语言中用于封装的程序单元
14. 类的静态方法和属性可以通过类本身和类的实例访问
15. 一个Java类可以实现多个接口
16. 一个Java类可以从多个基类派生
17. 关于Java程序中的对象，不正确的说法是（ B ）
18. 对象的状态通过相应的方法来获取和改变
19. 对象的删除通过Delete来实现
20. this代表当前对象
21. new操作通过调用构造函数创建相应的对象
22. 关于Java中的数组，下列叙述中正确的是（D）
23. 一个数组中可以包含不同类型的数据
24. 数组的下标从1开始
25. Java程序中可以定义变长数组
26. 数组可以静态初始化也可以动态初始化
27. 关于类的静态成员，下列说法中不正确的是（A）
28. 静态成员只能通过类访问，不能通过类的实例访问
29. 静态变量是全局变量
30. 所有Java程序中的main() 方法都是静态方法
31. 静态成员是指独立于对象的类成员
32. 关于方法的重载与覆盖，不正确的是（A）
33. 可以通过返回的类型不同来重载类的方法
34. 在一个类中，可以按重载要求对一个方法进行多次重载
35. 子类可以在覆盖父类某个方法的同时，对这个方法进行重载
36. 一个类的构造函数也可以重载多次
37. 关于线程不正确的是（C）
38. 线程机制提供了一种更高效的程序并行执行手段
39. 线程是一个程序进程中的一个执行分支
40. 一个Java进程的多个线程有各自的代码区和数据区
41. Java语言对多线程编程提供了良好的支持
42. 关于Java中的异常，不正确的是（A）
43. 所有异常都必须捕获并处理
44. 异常是程序运行时的非正常情况
45. Java程序的健壮性得益于Java系统的异常处理机制
46. 在Java语言中，异常是作为一种对象来处理的
47. 关于图形容器不正确的是 （C）
48. 图形界面中的组件通过容器来组织
49. 中间容器用来分组管理界面中的基本组件
50. 每个图形界面最多只有一个顶层容器
51. 所有的容器中都可以加入基本组件
52. 填空题（本大题共15小题，每小题2分，共30分）
53. Java类的三种显示访问控制符分别为\_\_public\_\_、\_protected\_与\_private\_。
54. Java源程序通过编译器\_Javac\_\_编译为\_字节\_码文件；
55. 编译后的目标Java程序在\_\_jvm\_\_上\_\_解释\_\_执行。
56. Java中的一个普通整型、字符数据和字节数据在机器中分别占\_\_4\_\_\_\_位、\_\_\_2\_\_\_\_位与\_\_\_1\_\_位。
57. 对象的状态由\_\_\_\_\_\_\_\_表现，状态的建立和改变由\_\_\_相应的方法\_实现。
58. Applet程序是嵌入在\_\_\_HTML\_\_\_中的小程序，而且是直接在 \_\_浏览器\_\_端执行的。
59. 在Java语言中，图形用户界面GUI组件可分为\_基本\_\_组件和\_容器\_\_组件。
60. Java语言系统内置的3个可以直接使用的流对象分别是\_System.in\_\_、\_System.out\_\_、\_System.err\_。
61. Java程序中的所有线程类都必须实现接口\_Runnable\_­­­­­­­\_，其中的方法\_run()\_­­­­­­­\_\_决定了线程的执行行为。
62. Java程序中抛出异常的关键字为\_throw\_，捕获异常的关键字为\_catch\_。
63. 程序填空题

(1) 以下程序需要引入java.util包中所有的类，请在下划线上填入相应的关键字或符号

\_import\_ java.util. \_Date\_

class DateDemo {

Date d;

public void f() {....}

}

(2) 补充下列Rect类中的空缺代码

public class Rect {

int w, h;

public Rect (int w, int h) {

\_\_this.w\_\_\_\_=\_\_\_w\_\_\_;

\_\_\_this.h\_\_\_=\_\_\_h\_\_\_;

}

public String getW( ) { \_\_\_return this.w\_\_\_\_\_\_\_\_\_ ; }

public String getH( ) { \_\_\_\_return this.h\_\_\_\_\_\_\_\_ ; }

public void setW(int w) { \_\_\_\_this.w = w\_\_\_\_\_\_; }

public void setH (int h) { \_\_\_\_\_this.h = h\_\_\_\_\_\_\_; }

}

(3) 对于以下Applet程序和相应的HTML文档，补充缺少的部分

public class AppletOne extends \_JApplet\_ {

public void paint(\_Graphics\_ g) {

g.drawString("Applet Demo",10,10);

}

}

<HTML>

<APPLET CODE=“\_AppletOne.class\_” WIDTH=“100” HEIGHT=“200”>

</ APPLET>

</HTML>

(4) public class IntParse {

public static void main(String[] args) {

String s="150";

long l=new Long(s). \_longValue\_ ( );

int i=Integer. \_parseInt\_ (s);

}

}

(5)  public class JFPDemo{

public static void main(String[] args){

JFrame jf=new JFrame("GUI ONE");

JPanel p=new JPanel();

JLabel l1=new JLabel("Label one");

JLabel l2=new JLabel(" Label two");

p.add(l2);

Container c=jf.\_getContentPane\_();

c.add(p,"North");

c.add(l1,"Center");

jf.setSize(100,200);

jf.set\_Visible\_(true);

}

}

三、简答题与程序分析（本大题共5小题，每小题 5分， 共25分）

1. Java语言的特点有哪些？

答:java语言吸收了[C++](https://baike.baidu.com/item/C++" \t "/Users/edz/Documents\\x/_blank)语言的各种优点，但是摒弃了C++里难以理解的[多继承](https://baike.baidu.com/item/%E5%A4%9A%E7%BB%A7%E6%89%BF" \t "/Users/edz/Documents\\x/_blank)、[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88/2878304" \t "/Users/edz/Documents\\x/_blank)等概念.并且最重要的是java提供了自动垃圾回收机制,不需要软件开发人员关注垃圾回收方面的问题.

java语言是一种面向对象的语言,这表示软件开发人员不需要严格通过过程思考,只需要关注当前对象行为和属性的封装

java语言可移植性高,一处开发,处处运行,只要部署好java运行环境,不论它运行在何种cpu、操作系统或者java编译器器上都产生一样的结果.

Java是多线程语言，它提供支持多线程的执行，通过线程能处理不同任务.

1. 分别解释进程与线程及其相互之间的关系。

答:进程：指在系统中能独立运行并作为资源分配的基本单位，它是由一组机器指令、数据和堆栈等组成的，是一个能独立运行的活动实体。

线程：线程是进程中的一个实体，作为系统调度和分派的基本单位。

从一定意义上讲，进程就是一个应用程序在处理机上的一次执行过程，它是一个动态的概念，而线程是进程中的一部分，进程包含多个线程在运行。

在引入线程的操作系统中，通常都是把进程作为分配资源的基本单位，而把线程作为独立运行和独立调度的基本单位。由于线程比进程更小，基本上不拥有系统资源，故对它的调度所付出的开销就会小得多，能更高效的提高系统内多个程序间并发执行的程度。

1. 图形界面中界面对象的布局有哪几种？

答:有五种

BorderLayout 边界布局管理器

FlowLayout 流式布局管理器

GridLayout网格布局管理器

GridBagLayout网格包布局管理器

CardLayout 卡片布局管理器

1. 解释Java程序中的异常机制。

答:异常指的是，那些在程序运行过程中，中断正常的指令流程的事件。在java程序设计语言中,所有的异常对象都是派生于Throwable类，

一般情况下，如果java中内置的异常类不能满足需求，可以自定义异常类只需要继承与Throwable类即可

Throwable有两个重要的子类：Exception（异常）和 Error（错误），二者都是 Java 异常处理的重要子类，各自都包含大量子类

Error（错误）:是程序无法处理的错误，表示运行应用程序中较严重问题。大多数错误与代码编写者执行的操作无关，而表示代码运行时 JVM（Java 虚拟机）出现的问题

Exception 类有一个重要的子类 RuntimeException。RuntimeException 类及其子类表示“JVM 常用操作”引发的错误

java异常可以使用 try包裹可能出现异常的代码 使用catch处理异常情况.

1. 分析以下程序，当分别输入数字1、3和字符’Y’时的执行结果，并解释其原因。

public class Exam{

public static void main(String[] args) {

int i=0;

String[] ch={"A","B","C"};

Scanner cin =new Scanner(System.in);

try {

System.out.println("----F1 ----");

System.out.println("请输入你的选项（1，2，3）");

i=cin.nextInt();

System.out.println(ch[i-1]);

}

catch(InputMismatchException e) {

System.out.println("----F2 ----");

System.out.println("输入数据格式错，要求是整型！");

}

catch(ArrayIndexOutOfBoundsException e) {

System.out.println("----F3 ----");

System.out.println("数组越界！");

}

finally {

System.out.println("----F4 ----");

System.out.println("执行完毕！");

}

}

}

答: 程序启动,scanner等待用户输入,System输出提示信息----F1 ---- 和 请输入你的选项（1，2，3）;nextInt方法获取用户输入;当用户输入1时,输出ch中下标为1-1的值 A;当输入3时,输出ch中下标为3-1的值 C;当输入Y时,用户输入的不是一个int值,程序发生InputMismatchException异常,输出----F2 ---- 和 输入数据格式错，要求是整型！;最后运行finally代码块,输出----F4 ---- 和 执行完毕！程序运行结束

四、编程题 （每小题7分，计35分）

* 1. 编写循环程序打印如图所示的X图案（要求使用循环，否则无效）。

V V

V V

V V

V V

V

V V

V V

V V

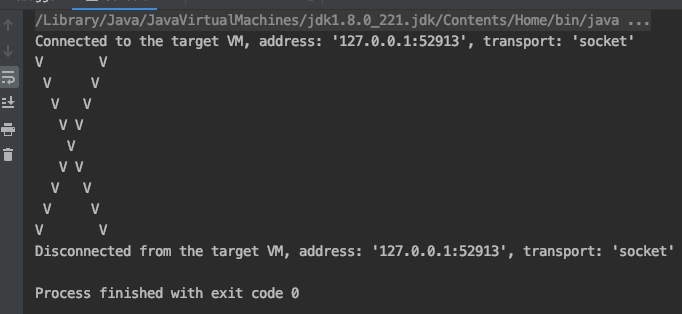
V V

答:

代码片段:

|  |
| --- |
| package com.joker.donghua;  /\*\*  \* 编写循环程序打印如图所示的X图案 要求使用循环，否则无效  \*/  public class PrintX {  public static void main(String[] args) {  int layer = 9;  for (int x = 0; x < layer; x++) {  for (int y = 0; y < layer; y++) {  if (y == x || y == layer - 1 - x) {  System.out.print("V");  } else {  System.out.print(" ");  }  }  System.out.println();  }  }  } |

运行截图:



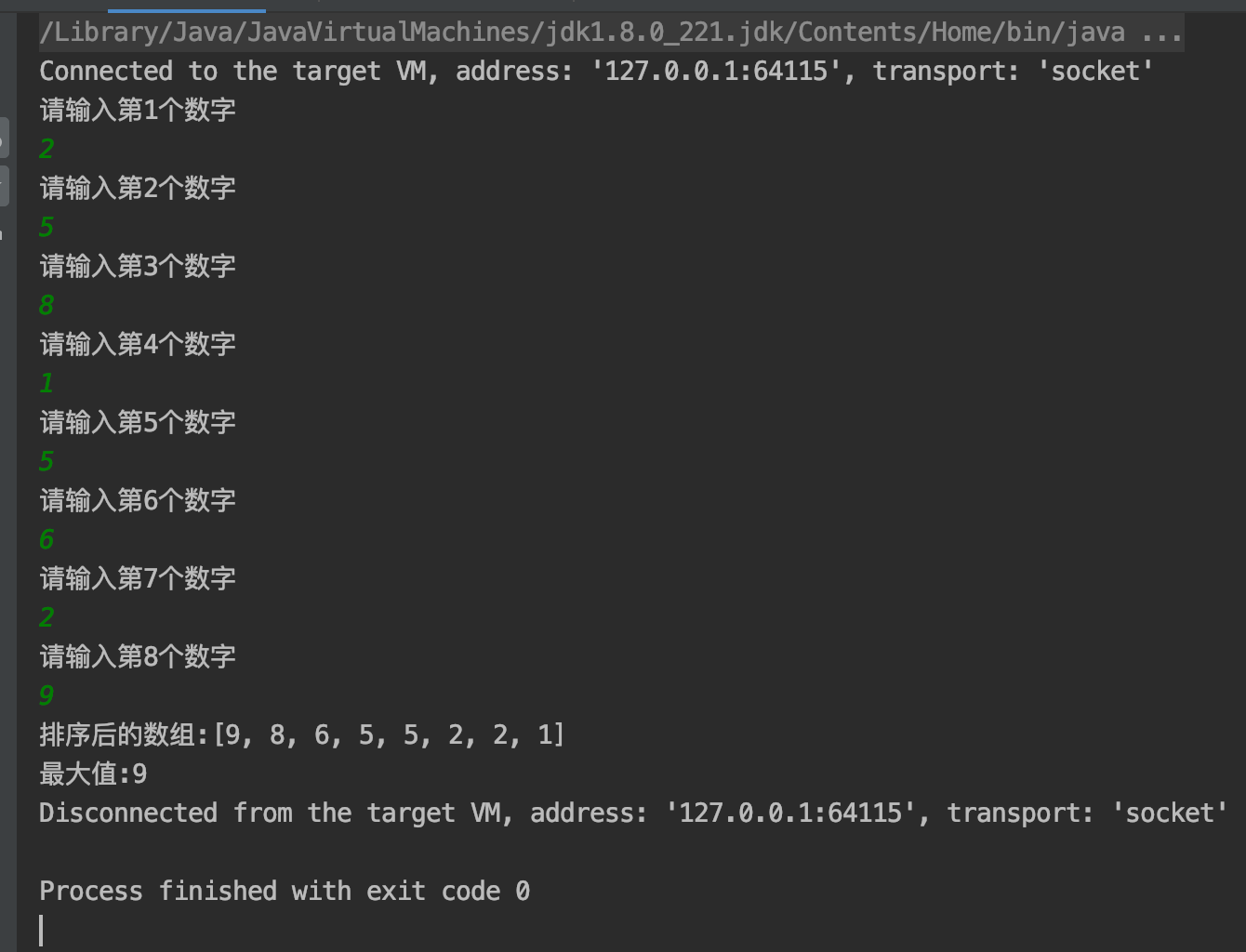
* 1. 编写程序MaxNum，接收键盘输入一系列整数（不超过8个），将它们存入数组A，然后按降序排序，最后显示排序后的数组及最大值。

答:

MaxNum.class

|  |
| --- |
| package com.joker.donghua;  import java.util.Arrays;  import java.util.Collections;  import java.util.Scanner;  /\*\*  \* 编写程序MaxNum，接收键盘输入一系列整数（不超过8个），  \* 将它们存入数组A，然后按降序排序，最后显示排序后的数组及最大值。  \*  \*/  public class MaxNum {  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  Integer maxNum = 0;  Integer[] A = new Integer[8];  for (int x = 0; x < 8; x++) {  System.out.println("请输入第" + (x + 1) + "个数字");  int i = in.nextInt();  A[x] = i;  if (maxNum < i) {  maxNum = i;  }  }  Arrays.sort(A, Collections.reverseOrder());  System.out.println("排序后的数组:" + Arrays.toString(A));  System.out.println("最大值:" + maxNum);  }  } |

运行结果截图:



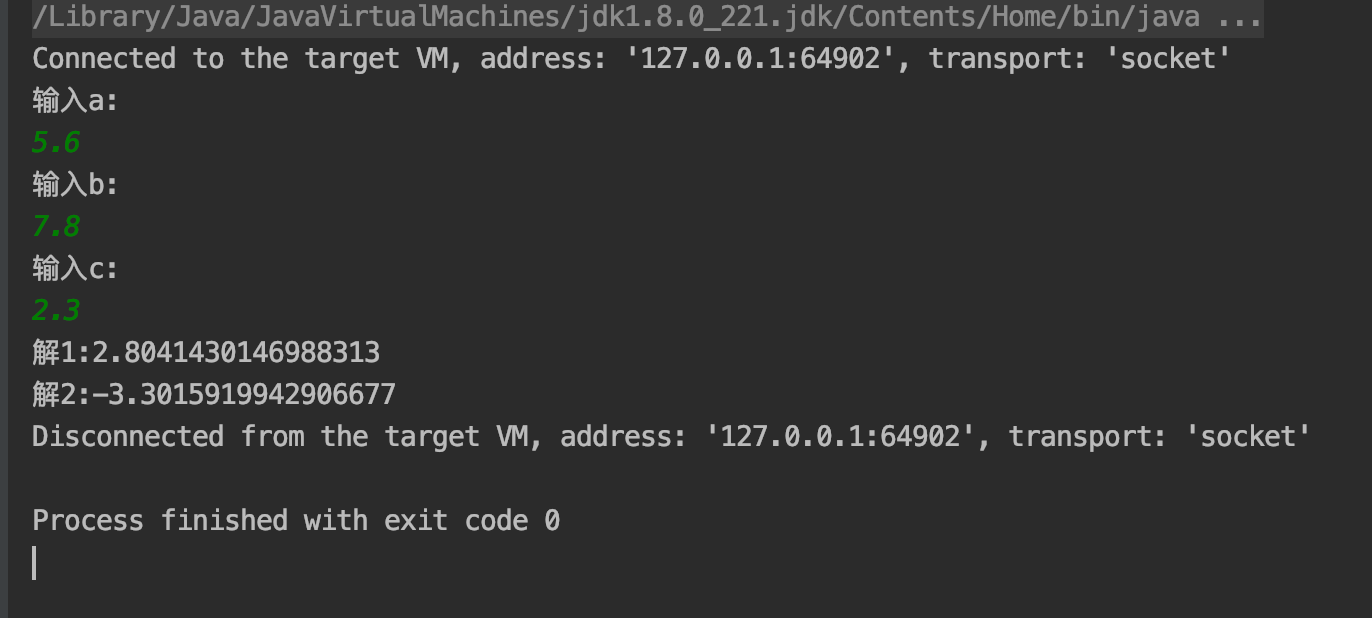
* 1. 设计一个实数域上二次方程的解方案：首先定义一个方程类Equ，使得方程对象包含三个实系数属性A，B和C，传输三个参数可构造方程对象，其解方法answer()给出方程在实数域中的完整解决方案(即各种情况下的答案)。最后在主函数中定义各种情况的方程对象实例并显示方程属性和答案。

答:

Equ.class

|  |
| --- |
| package com.joker.donghua;  import java.util.Scanner;  /\*\*  \* 设计一个实数域上二次方程的解方案：首先定义一个方程类 Equ，  \* 使得方程对象包含三个实系数属性A，B和C，传输三个参数可构造方程对象，  \* 其解方法answer()给出方程在实数域中的完整解决方案(即各种情况下的答案)。  \* 最后在主函数中定义各种情况的方程对象实例并显示方程属性和答案  \*  \*/  public class Equ {  private double A;  private double B;  private double C;  public Equ(double A, double B, double C) {  this.A = A;  this.B = B;  this.C = C;  }  private void answer() {  // b^2-4ac  double delta = Math.pow(this.B, 2) - 4 \* this.A \* this.C;  if (delta < 0) {  System.out.println("无实数解");  return;  }  double result1 = -this.B / (A \* this.A) + Math.sqrt(delta);  double result2 = -this.B / (A \* this.A) - Math.sqrt(delta);  System.out.println("解1:" + result1);  System.out.println("解2:" + result2);  }  public static void main(String[] args) {  double A, B, C;  Scanner scanner = new Scanner(System.in);  System.out.println("输入a:");  A = scanner.nextDouble();  System.out.println("输入b:");  B = scanner.nextDouble();  System.out.println("输入c:");  C = scanner.nextDouble();  Equ instance = new Equ(A, B, C);  instance.answer();  }  } |

运行结果截图:



* 1. 开发一个完整的Applet程序，以坐标点 (1，5)为左上角，分别以3和5为宽和高画一个矩形Rec，并嵌入到一个HTML文档Rec.html中，然后结合本例说明开发一个Applet程序的过程。

答:

1)建立Java Applet源程序。

2)把Applet的源程序转换为字节码文件。

3)编制使用class 的HTML文件。在HTML文件内放入必要的<APPLET>语句

代码片段:

Rec.java

|  |
| --- |
| package donghua;  import java.applet.Applet;  import java.awt.\*;  /\*\*  \* 以坐标点 (1,5)为左上角绘制宽为3，高为5的矩形  \*  \*/  public class Rec extends Applet {  @Override  public void paint(Graphics g) {  g.drawRect(1, 5, 3, 5);  }  } |

Rec.html

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>applet-曹会金</title>  </head>  <body>  <applet code="donghua.Rec.class" width="100" height="100" codebase=".."></applet>  </body>  </html> |

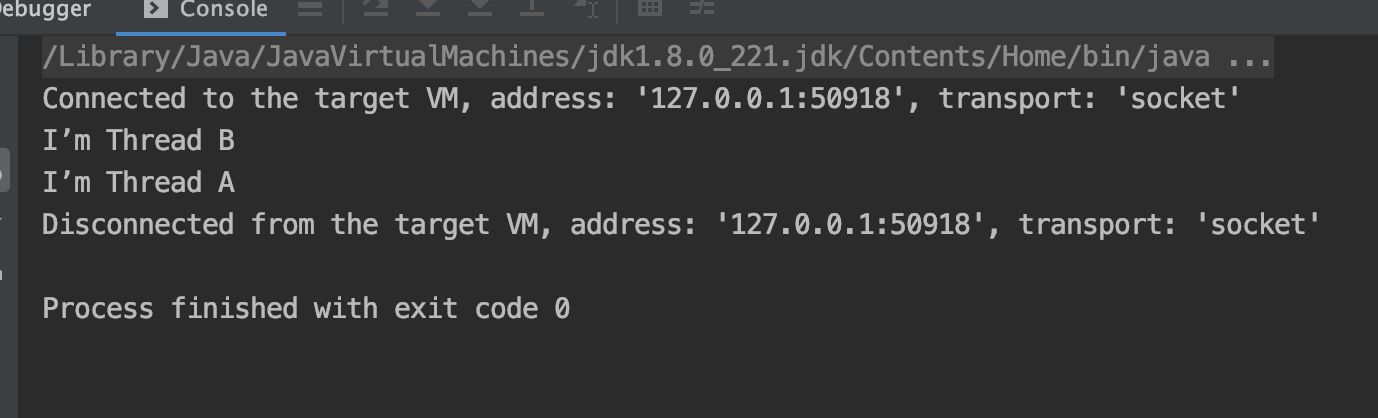
程序运行截图:



* 1. 定义两个线程类You和Me，执行线程时分别显示“I’m Thread A”和“I’m Thread B”， 然后在main()函数中创建两个线程的对象，并分别启动两个线程。

答:

程序运行截图:



代码片段:

You.java

|  |
| --- |
| package com.joker.donghua;  /\*\*  \* 线程You 输出I’m Thread A  \*/  public class You extends Thread {  @Override  public void run() {  super.run();  System.out.println("I’m Thread A");  }  } |

Me.java

|  |
| --- |
| package com.joker.donghua;  /\*\*  \* 线程A 输出I’m Thread B  \*/  public class Me extends Thread {  @Override  public void run() {  super.run();  System.out.println("I’m Thread B");  }  } |

ThreadTest.java

|  |
| --- |
| package com.joker.donghua;  /\*\*  \* 多线程测试类  \*/  public class ThreadTest {  public static void main(String[] args) {  You you = new You();  Me me = new Me();  you.start();  me.start();  }  } |