



Java 程序设计

从入门到精通

毛雪涛/丁毓峰 著

从编程菜鸟，到Java语言大神的实用宝典！

- 资历深：凝聚笔者10多年的开发经验
- 内容全：囊括了Java的150多个知识点
- 实例多：书中包括100多个实例
- 技术新：结合当下Java的各种

280道Java面试题

460种JavaScript特效源代码

100个HTML标签的详细用法

136个CSS标签的详细用法

SQL结构化查询语言使用大全

300道Java课后练习题

20天横扫Java基础视频，20小时

Java基础实战——Bank项目，20小时

Oracle、SQL、PLSQL 视频教程，20小时

JDBC视频教程，20小时

Java 9新特性视频教程，10小时

Java——JUC视频教程，10小时

Java——NIO视频教程，10小时

JavaWeb视频基础，10小时

JavaWeb项目实战，10小时

赠

电子书

赠

视频



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

下册 高级应用篇

Java 程序设计

从入门到精通

毛雪涛/丁毓峰 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以基础知识与实际动手编码相结合为原则,以 Java 从基础到核心开发为主线,通过 Java 开发中的基本知识和常见的应用技术,介绍了实用性极强且频繁使用的 Java 技术。

本书共 4 篇 18 章,分上、下两册,涵盖的主要内容有:Java 理论基础、Java 开发环境、Java 语言基础、流程控制、字符串、数组、对象与类、继承和多态、JSP 技术、多线程技术、Java 异常处理和调试技术、I/O 系统编程、Java 数据库编程、Java 网络编程、Java 图形界面编程及两个综合案例。

本书内容丰富、实例典型、实用性强,适合各个层次想要学习 Java 开发技术的读者阅读,尤其适合没有 Java 基础而又要快速成长为 Java 程序员的读者阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计从入门到精通 / 毛雪涛, 丁毓峰著. —北京: 电子工业出版社, 2018.1

ISBN 978-7-121-32706-3

I. ①J… II. ①毛… ②丁… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 226584 号

策划编辑: 徐 岩

责任编辑: 张 毅

特约编辑: 赵树刚

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×1092 1/16 印张: 37.75 字数: 966 千字

版 次: 2018 年 1 月第 1 版

印 次: 2018 年 1 月第 1 次印刷

定 价: 88.00 元 (共 2 册)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

本书咨询联系方式: (010) 57565805。

前言

随着信息技术的飞速发展，对软件开发的要求越来越高，软件开发人才的需求市场热度不减。Java 语言在快速开发、可扩展性、平台兼容性等方面具有得天独厚的优势。特别是 Java 面向对象的特点，使得 Java 的解决方案在各行各业得到了广泛的应用。程序员要想进入 Java 开发行业，不仅需要有扎实的 Java 语言基础，而且要深入理解 Java 的理论基础，最好还要熟悉应用开发中有典型意义和实用价值的各类开发实例及案例，这样才能在就业严峻的市场环境中有较强的职场竞争力和职业前景。

目前图书市场上关于 Java 开发及框架整合的图书琳琅满目，但 Java 语言发展迅速，真正既透彻易懂又紧密联系实际的、既结合最新技术发展又指导编程演练的图书却寥寥无几。本书从理论到实践，通过总结归纳 Java 语言中的核心知识，结合实例训练，再加上项目案例进行练习拓展，通过各种典型模块和项目案例来指导读者从零基础开始到能够应用开发，让读者全面、深入、透彻地理解 Java 开发的基础知识和各种主流技术，逐步将读者打造成具有实际开发水平和项目实战能力的 Java 程序员。

本书有何特色

1. 讲解特色

为了让读者更快地上手，本书特别设计了适合初学者的学习方式，用准确的语言总结概念、用直观的图示演示过程、用详细的注释解释代码、用形象的比喻帮助记忆。

- 知识点介绍：在每节的开始或者每个概念的讲解开始讲解相关知识点，让零基础读者能了解概念、顺利入门。
- 实例：为每个实例进行编号，便于检索和循序渐进地学习。
- 实例代码：与实例对应，层次清楚，语句简洁，注释丰富。
- 运行结果：对实例的运行结果给出图示，直观清楚。
- 代码解析：针对实例的代码和关键点，逐一解释，让读者能融会贯通。
- 贴心提示：全书关键点都给出了注意和提示，让读者能举一反三。

2. 内容特色

- 经验丰富：笔者有 10 多年的编程经验，熟悉 Java 的各项知识细节。
- 只讲重点：Java 涉及的知识点众多，本书讲解了常用的 300 多个知识点，其他内容可以参考本书的赠送资料。
- 实例众多：针对每个知识点都给出了典型的实例程序，让读者边学边练习。

本书内容及知识体系

第 1 篇 入门篇（第 1~4 章）

本篇介绍了 Java 开发环境的配置和基础知识，主要包括认识 Java、搭建 Java 开发环境、Java 语言基础、流程控制等内容。

第 2 篇 高级技术篇（第 5~10 章）

本篇介绍了 Java 开发中最常遇到的高级技术问题，主要包括字符串、数组、对象与类、Java 中的继承、多态与抽象类、Java Web 应用——JSP 技术等内容。

第 3 篇 核心技术篇（第 11~16 章）

本篇主要介绍了 Java 实际应用中最为常用到的核心技术，主要包括多线程、异常处理、I/O 系统、Java 数据库编程、Java 网络编程和 Java 图形界面编程等内容。

第 4 篇 实战篇（第 17、18 章）

本篇主要介绍了两个综合开发实例。实例涵盖了系统开发从需求分析、数据库设计到界面开发、业务逻辑开发等全部过程。

适合阅读本书的读者群

- ☐ 想从事编程工作的人员。
- ☐ 大中专院校的学生。
- ☐ 使用 Java 作为开发语言的公司。
- ☐ 编程爱好者。
- ☐ 参加实习的“菜鸟”程序员。
- ☐ 程序测试及维护人员。

目 录

第 3 篇 核心技术篇

第 11 章 多线程.....	311
11.1 线程基础.....	311
11.1.1 什么是线程.....	312
11.1.2 Thread 类创建线程的方法.....	312
11.1.3 Thread 类创建线程的步骤.....	315
11.1.4 Runnable 接口创建线程的方法.....	316
11.1.5 Runnable 接口创建线程的步骤.....	318
11.2 线程的生命周期.....	319
11.2.1 线程的 4 种状态.....	320
11.2.2 线程的创建和启动.....	321
11.3 线程调度.....	322
11.4 线程同步.....	325
11.4.1 同步的概念.....	325
11.4.2 同步的格式.....	327
11.4.3 同步的应用.....	328
11.5 线程通信.....	330
11.5.1 生产者/消费者.....	330
11.5.2 共享队列.....	332
11.5.3 运行生产者/消费者.....	332
11.6 死锁.....	335
11.7 拓展训练.....	337
11.7.1 训练一：篮球运动员的比赛安排.....	337
11.7.2 训练二：ATM 的实现.....	339

11.8 技术解惑.....	343
11.8.1 线程有哪 5 种基本状态，它们如何相互转换	343
11.8.2 如何让一个线程运行特定的时间后优雅地终止	343
第 12 章 异常处理	345
12.1 异常的概念.....	345
12.2 异常的基本样式.....	346
12.3 Java 异常类	347
12.3.1 异常类的层次结构	348
12.3.2 异常处理方法	349
12.4 异常捕获.....	351
12.4.1 异常捕获处理语句块	351
12.4.2 必须执行语句块	354
12.4.3 必须执行语句块的意外	355
12.5 异常抛出.....	356
12.5.1 异常抛出的概念	356
12.5.2 throw	357
12.5.3 throws	358
12.5.4 抛出异常的步骤	361
12.6 自定义异常.....	362
12.6.1 创建用户自定义异常	362
12.6.2 捕获运行时异常	364
12.7 拓展训练.....	365
12.7.1 训练一：强制类型转换异常	365
12.7.2 训练二：找不到指定类时异常	366
12.8 技术解惑.....	367
12.8.1 如何优雅地处理异常	367
12.8.2 详解 throw 与 throws 关键字的用法区别	368
第 13 章 I/O 系统	370
13.1 I/O 概述	370
13.2 文件.....	371
13.2.1 File 类.....	371
13.2.2 File 类的应用	373

13.2.3	RandomAccessFile 类	375
13.2.4	RandomAccessFile 类的应用	376
13.3	字节流 InputStream、OutputStream	379
13.3.1	字节输入、输出流	379
13.3.2	字节文件输入、输出流	381
13.3.3	字节缓冲输入、输出流	383
13.3.4	字节数据输入、输出流	384
13.3.5	字节对象输入、输出流	386
13.4	字符流 Reader、Writer	390
13.4.1	字符读、写流	390
13.4.2	字符输入、输出流	391
13.4.3	文件读、写字符流	393
13.4.4	字符缓冲区读、写流	394
13.5	拓展训练	396
13.5.1	训练一：按顺序创建文件	396
13.5.2	训练二：将一个大文件分割为多个小文件	397
13.5.3	训练三：生成 ZIP 压缩文件	399
13.6	技术解惑	402
13.6.1	把 InputStream 转换成 String 的几种方法	402
13.6.2	读取大文件用哪个类合适	403
第 14 章	Java 数据库编程	405
14.1	数据库简介	405
14.1.1	数据库的概念	405
14.1.2	数据库的特点	406
14.2	SQL 语句	406
14.2.1	SQL 语句的分类	407
14.2.2	SELECT 查询语句	407
14.2.3	INSERT 插入语句	407
14.2.4	UPDATE 更新语句	408
14.2.5	DELETE 删除语句	408
14.2.6	CREATE 建表语句	408
14.2.7	ALTER 修改表语句	409
14.2.8	DROP 删除表语句	409

14.3	MySQL	409
14.3.1	MySQL 的安裝配置	410
14.3.2	MySQL 的使用	413
14.4	JDBC	416
14.4.1	JDBC 简介	416
14.4.2	JDBC 常见接口和类	417
14.4.3	JDBC 数据库连接步骤	417
14.4.4	加载 JDBC 驱动程序	418
14.4.5	获取 Connection 连接对象	421
14.4.6	创建 Statement 对象	423
14.4.7	创建 PreparedStatement 对象	425
14.4.8	创建 ResultSet 对象	427
14.5	执行 SQL 语句的方式	430
14.5.1	使用 JDBC API	430
14.5.2	使用 execute()方法执行 SQL 语句	430
14.6	Java 与各种数据库的连接	431
14.7	数据库开发案例	433
14.7.1	构造方法	433
14.7.2	数据查询	434
14.7.3	主方法	436
14.8	拓展训练	437
14.8.1	训练一：向数据库中批量插入数据	437
14.8.2	训练二：将图片文件存入数据库	439
14.9	技术解惑	441
14.9.1	多个 Connection 能实现事务吗	441
14.9.2	如何测试 java.sql.Connection 类当前的连接是否正常可用	442
第 15 章	Java 网络编程	443
15.1	网络程序设计基础	443
15.1.1	局域网与因特网	443
15.1.2	网络协议	444
15.1.3	端口和套接字	444
15.2	TCP 程序设计基础	446
15.2.1	InetAddress 类	446

15.2.2	ServerSocket 类.....	448
15.2.3	TCP 网络程序.....	450
15.3	UDP 程序设计基础.....	453
15.3.1	DatagramPacket 类	453
15.3.2	DatagramSocket 类	453
15.3.3	UDP 网络程序.....	454
15.4	拓展训练.....	457
15.4.1	训练一：实现网络文件的下载.....	457
15.4.2	训练二：模仿 QQ 创建实时聊天室	459
15.5	技术解惑.....	463
15.5.1	动态替换 InetAddress 中 DNS 的做法.....	463
15.5.2	输出 datagramPacket 的字符串时后面有很多小方块如何解决	464
第 16 章	Java 图形界面编程	465
16.1	AWT 简介	465
16.2	Swing 基础	466
16.2.1	Swing 的类层次结构.....	467
16.2.2	Swing 的特点.....	468
16.2.3	Swing 程序结构简介.....	468
16.3	事件类.....	470
16.3.1	事件类分类.....	470
16.3.2	动作事件类.....	471
16.3.3	调整事件类.....	471
16.3.4	焦点事件类.....	472
16.3.5	项目事件类.....	472
16.3.6	键盘事件类.....	473
16.3.7	鼠标事件类.....	473
16.3.8	窗口事件类.....	474
16.4	事件监听器.....	475
16.4.1	事件监听器接口.....	475
16.4.2	事件监听器的特点.....	477
16.5	轻量级容器.....	477
16.5.1	根面板 (JRootPane)	478
16.5.2	面板 (JPanel)	478

16.5.3	分层面板 (JLayeredPane)	479
16.5.4	滚动窗口 (JScrollPane)	479
16.5.5	分隔板 (JSplitPane)	479
16.5.6	选项板 (JTabbedPane)	480
16.5.7	工具栏 (JToolBar)	480
16.6	Swing 组件	480
16.6.1	按钮 (JButton)	481
16.6.2	复选框 (JCheckBox)	481
16.6.3	单选框 (JRadioButton)	484
16.6.4	组合框 (JComboBox)	485
16.6.5	进度条 (JProgressBar)	485
16.6.6	表格 (JTable)	485
16.6.7	树 (JTree)	486
16.7	拓展训练	487
16.7.1	训练一：设置闪烁的标题栏	487
16.7.2	训练二：建立简单的时钟布局	489
16.7.3	训练三：给你最喜爱的脱口秀演员投票	492
16.7.4	训练四：包含图片的弹出菜单	495
16.7.5	训练五：震动效果的提示信息	497
16.8	技术解惑	499
16.8.1	Swing 重要吗	499
16.8.2	Swing 是否适合做大型项目及桌面应用	499

第4篇 实战篇

第17章	综合案例：学生信息管理系统	501
17.1	需求分析	501
17.2	系统设计	501
17.3	登录模块	503
17.3.1	设计构想	503
17.3.2	实现登录模块	503
17.4	学生信息管理系统	506
17.4.1	设计构想	506

17.4.2	实现学生信息管理系统主界面	507
17.4.3	增加学生信息	509
17.4.4	删除学生信息	512
17.4.5	修改学生信息	515
17.4.6	查询学生信息	519
17.4.7	成绩查询	523
17.4.8	退出系统	526
17.5	教师信息管理系统	528
17.5.1	教师信息管理系统主界面	528
17.5.2	增加教师信息	531
17.5.3	删除教师信息	534
17.5.4	修改教师信息	537
17.5.5	录入成绩	541
17.5.6	删除成绩	544
17.5.7	修改成绩	547
17.5.8	教师信息查询	550
17.5.9	退出系统	554
17.6	开发常见问题与解决	554
第 18 章	综合案例：进销存管理系统	556
18.1	系统分析	556
18.2	系统设计	557
18.2.1	系统架构	558
18.2.2	关键数据结构图	559
18.3	开发环境	559
18.3.1	服务器软件环境	559
18.3.2	服务器硬件环境	560
18.4	数据表设计	560
18.4.1	订单表设计	560
18.4.2	商品表设计	560
18.4.3	用户数据表设计	561
18.4.4	货架数据表设计	561
18.4.5	销售记录数据表设计	561
18.4.6	工作记录数据表设计	561

18.5	创建项目.....	562
18.6	系统文件夹组织结构.....	562
18.7	公共类设计.....	563
18.8	登录模块设计.....	565
18.9	系统主窗口设计.....	565
18.10	进货单模块设计.....	569
18.11	销售单模块设计.....	571
18.12	库存管理模块设计.....	573
18.13	数据库备份与恢复模块设计.....	575
18.14	运行项目.....	576
18.15	系统打包发行.....	576

第3篇 核心技术篇

第 11 章

多线程

我们在饭店聚餐时，多人同时吃一道菜的时候很容易发生争抢。例如，上了一道好菜，两人同时夹这道菜，一人刚伸出筷子，结果伸到的时候菜已经被夹走了……怎么办呢？此时必须等一人夹完一口菜之后，另一人再夹菜。也就是说，资源共享就会发生争抢，这就是多线程争抢资源的问题。

线程是一个单独的程序流程。多线程是指一个程序可以同时运行多个任务，每个任务由一个单独的线程来完成。如果程序被设置为多线程，则可以提高程序运行的效率和处理速度。可以通过控制线程来控制程序的运行，如操作线程的等待、休眠、唤醒等。本章将向读者介绍 Java 中的多线程机制，以及如何通过 Java 语言进行多线程编程。

提示：多线程的概念读者开始觉得不太好理解，那就想想在生活中当资源共享出现冲突的时候该怎么办呢？除了上面说的抢菜，还有公交车上抢座、多人雨中争抢出租车等，都是这样的例子。线程包括线程周期、线程调度、线程同步、线程通信和死锁等概念，只有理解了这些概念，才能更好地在程序中使用线程。

11.1 线程基础

多个线程可以同时在一个程序中运行，并且每个线程完成不同的任务。Java 中线程的实现通常有两种方法：派生 Thread 类和实现 Runnable 接口。本节主要讲述线程的概念和创建线程的方法。

11.1.1 什么是线程

传统的程序设计语言同一时刻只能执行单任务，效率非常低下，如果网络程序在接收数据时发生阻塞，则只能等到程序接收完数据之后才能继续运行。随着 Internet 的飞速发展，这种单任务运行的状况越来越不被接受。如果网络程序在接收数据时发生阻塞，那么后台服务程序就会一直处于等待状态而不能继续任何操作；如果这种阻塞情况经常发生，那么 CPU 资源将完全处于闲置状态。

多线程实现后台服务程序可以同时处理多个任务，并且不会发生阻塞现象。多线程是 Java 语言的一个很重要的特征。多线程程序设计最大的特点就是能够提高程序的执行效率和处理速度。Java 程序可同时运行多个相对独立的线程。例如，在开发一个 E-mail 电邮系统时，通常需要创建一个线程来接收数据，创建另一个线程来发送数据，即使发送数据线程发生阻塞，接收数据线程仍然可以运行。

线程 (Thread) 是控制线程 (Thread of Control) 的缩写，它是具有一定顺序的指令序列 (所编写的程序代码)、存放方法中定义局部变量的栈和一些共享数据。线程是相互独立的，每个方法的局部变量和其他线程的局部变量是分开的，因此，任何线程都不能访问除自身之外的其他线程的局部变量。如果两个线程同时访问同一个方法，那么每个线程将各自得到此方法的一个拷贝。

Java 提供的多线程机制使得一个程序可同时执行多个任务。线程有时也被称为小进程，它是从一个大进程里分离出来的小的独立的线程。由于实现了多线程技术，Java 显得更健壮。多线程带来的好处是更好的交互性能和实时控制性能。多线程是强大而灵巧的编程工具，但要用好它却不是一件容易的事。

警告：多线程程序如果使用不当，则可能会导致一系列错误，并且不容易排查错误，所以一定要在掌握其方法和特性后再使用。

在多线程编程中，每个线程都通过代码实现线程的行为，并将数据提供给代码操作。代码和数据有时是相对独立的，可分别提供给线程。多个线程可以同时处理同一代码和同一数据，不同的线程也可以处理各自不同的代码和数据。

11.1.2 Thread 类创建线程的方法

了解了线程的概念后，现在来介绍创建线程的方法。Java 中有两种方法创建线程：一种是对 Thread 类进行派生并覆盖 run() 方法；另一种是通过实现 Runnable 接口创建。

本节先介绍如何通过 Thread 类创建线程，下一节将向读者介绍另一种方法。第一种派生方法如下：继承 Thread 类并覆盖 Thread 类的 run() 方法完成线程类的声明，通过 new 关键字创建派生线程类的线程对象。run() 方法中的代码实现了线程的行为。

前面的程序都是声明一个公共类，并在类内实现一个 main() 方法。事实上，前面这些程序就是一个单线程程序。当它执行完 main() 方法的程序后，线程正好退出，程序同时结束运行。下面给出一个系统创建单线程程序的例子。

【实例 11.1】创建单线程实例。

```

01 //11.1 OnlyThread.java           描述: 创建单线程
02 public class OnlyThread{
03     public static void main(String args[]){
04         run();                     //调用静态 run()方法
05     }
06     //实现 run()方法
07     public static void run()
08     {
09         for (int count = 1,row = 1; row < 10; row++,count++) //循环计算输出的*数量
10         {
11             for (int i = 0; i < count; i++) //循环输出指定变量 count 值的*数量
12             {
13                 System.out.print('*');    //输出*符号
14             }
15             System.out.println();        //输出换行符
16         }
17     }
18 }

```

【代码解析】

该程序只是建立了一个单一线程并执行的普通小程序，并没有涉及多线程的概念。该程序首先创建类 OnlyThread（第 2~5 行），然后创建一个 run()方法（第 7~17 行），在该方法中使用了嵌套 for 循环，外层 for 循环用来循环计算输出的*数目（第 9~16 行），内层 for 循环用来输出*符号（第 11~14 行），run()方法在 main()方法中调用（第 4 行）。

【运行结果】

在 Eclipse 项目资源管理中新建项目 Chapter11，在该项目的 src 文件夹下创建 example 包，本章的实例代码都在此包中创建。新建 OnlyThread.java 文件，在其 main()方法中输入上述代码。在 Eclipse 左侧的导航栏中选中 OnlyThread.java 文件，单击鼠标右键，在弹出的快捷菜单中选择“运行方式”|“Java 应用程序”命令，项目将自动编译运行 OnlyThread.java 的代码，运行结果如图 11.1 所示。

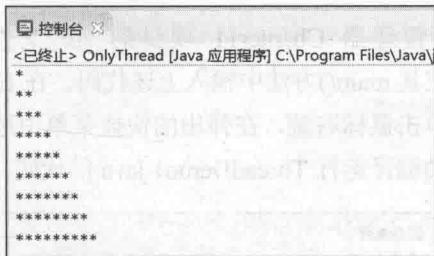


图 11.1 OnlyThread.java 运行结果

提示：java.lang.Thread 类是一个通用的线程类，由于默认情况下 run()方法是空的，直接通过 Thread 类实例化的线程对象不能完成任何事，所以可以通过派生 Thread 类，并用具体程序代码覆盖 Thread 类中的 run()方法，来实现具有各种不同功能的线程类。

在程序中创建新的线程的方法之一是继承 Thread 类，并通过 Thread 子类声明线程对象。

【实例 11.2】通过 Thread 类创建线程实例。

```

01 //程序 ThreadDemo1.java      描述: 创建一个新的线程
02 class ThreadDemo1 extends Thread{
03     //声明 ThreadDemo 构造方法
04     ThreadDemo1(){}
05     //声明 ThreadDemo 带参数的构造方法
06     ThreadDemo1(String szName)
07     {
08         super(szName);           //调用父类的构造方法
09     }
10     //重载 run() 方法
11     public void run()
12     {
13         for (int count = 1,row = 1; row < 10; row++,count++) //循环计算输出的*数量
14         {
15             for (int i = 0; i < count; i++) //循环输出指定变量 count 值的数量*
16             {
17                 System.out.print('*'); //输出*符号
18             }
19             System.out.println(); //输出换行符
20         }
21     }
22     public static void main(String argv[]){
23         ThreadDemo1 td = new ThreadDemo1(); //创建并初始化 ThreadDemo1 类型对象 td
24         td.start(); //调用 start() 方法执行一个新的线程
25     }
26 }

```

【代码解析】

该程序与实例 11.1 的运行结果相同，但是仔细对比发现，实例 11.1 中对 run() 方法的调用在实例 11.2 中变成了对 start() 方法的调用（第 22~25 行），并且实例 11.2 明确派生 Thread 类，创建新的线程类（第 2 行），实现了 Thread 类的 run() 方法（第 11~21 行）。其他输出*符号的方法与实例 11.1 一样，具体可以参考实例 11.1。

【运行结果】

在 Eclipse 项目资源管理器 Chapter11 项目的 src 文件夹的 example 包下新建 ThreadDemo1.java 文件，在其 main() 方法中输入上述代码。在 Eclipse 左侧的导航栏中选中 ThreadDemo1.java 文件，单击鼠标右键，在弹出的快捷菜单中选择“运行方式”|“Java 应用程序”命令，项目将自动编译运行 ThreadDemo1.java 的代码，运行结果如图 11.2 所示。

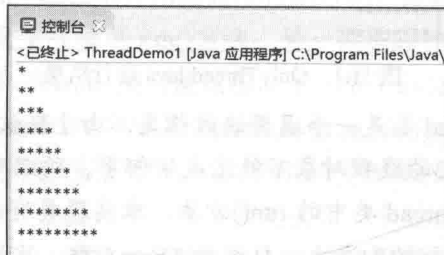


图 11.2 ThreadDemo1.java 运行结果