



高等学校计算机教材

Java

实用教程 (第3版)

◎ 郑阿奇 主编

◎ 姜乃松 殷红先 编著



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>

高等学校计算机教材

Java 实用教程

(第3版)

郑阿奇 主编
姜乃松 殷红先 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本教程以甲骨文最新发布的 Java 8 为基础,对《Java 实用教程》(第 2 版)进行了全新改版,内容循序渐进、深入浅出,精心设计每一个实例,结构安排更为合理,使读者准确把握 Java 的知识点。

教程包括 Java 实用教程、实验、习题和习题答案四部分(限于篇幅,习题和习题答案通过网络以电子文档形式提供)。本教程在讲解内容后紧跟实例,很多章节最后还配备有综合实例,对前面学习的主要知识进行综合应用。实验部分也是通过实例引导读者学习,并提出思考问题,最后在原来基础上让读者自己进行操作和编程练习。

全书以开源软件 Eclipse 作为 Java 的集成开发环境,使得编写、调试、运行 Java 程序变得更为简便。

本书既可作为大学本科和专科计算机及相关专业的教材,也可作为 Java 自学者或者应用开发者的参考书。

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 实用教程/郑阿奇主编;姜乃松,殷红先编著. —3 版. —北京:电子工业出版社,2015.9

高等学校计算机教材

ISBN 978-7-121-26622-5

I. ①J… II. ①郑… ②姜… ③殷… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 159854 号

策划编辑:程超群

责任编辑:左 雅

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:25.5 字数:734 千字

版 次:2005 年 5 月第 1 版

2015 年 9 月第 2 版

印 次:2015 年 9 月第 1 次印刷

印 数:3 000 册 定价:52.00 元

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。

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

服务热线:(010) 88258888。

前 言

Java 是原 Sun 公司（现已被甲骨文公司收购）开发的新一代编程语言，其简单、面向对象、多线程、跨平台等特性深受人们的欢迎。目前，学习 Java 已经成为一种时尚和潮流。

2005 年，为了满足教学和应用开发实践的要求，我们编写了《Java 实用教程》一书，把学习和应用相结合，收到了较好的效果，后又修订出版了第 2 版。

《Java 实用教程》（第 3 版）以 Java 最新的版本 Java 8 为基础，对第 2 版进行了全新改版，内容循序渐进、深入浅出，精心设计每一个示例，结构安排更为合理，使读者准确把握 Java 的知识点。实用教程一般在讲解一项内容后紧跟相关的实例演示，很多章节最后还配备了一个综合实例，使学习者综合应用已经学过的主要知识。实验部分也是通过实例先引导读者学习，再提出思考问题，最后在原来基础上让读者自己进行操作和练习编程。实例程序均在 JDK 8（Java SE Development Kit 8 Update 45）上调试通过。全书以开源软件 Eclipse 作为 Java 的集成开发环境，这使得编写、调试、运行 Java 程序变得更为简便。

实际上，本教程不仅适合教学，也非常适合 Java 的各类培训，以及使用 Eclipse 编程开发应用程序的用户学习和参考。只要阅读本书，结合上机操作进行练习，就能在较短的时间内基本掌握 Java 知识及其应用技术。

与本书配套的同步电子课件，可直接用于课堂教学。书中的源代码和电子课件，可从华信教育资源网（<http://www.hxedu.com.cn>）上免费下载。

本书由南京师范大学姜乃松和殷红先编写，南京师范大学郑阿奇任主编并对全书进行统稿、定稿。其他一些同志参加本书的基础工作，在此表示诚挚的谢意。

参加本套丛书（《高等学校计算机教材》）编写的还有梁敬东、陆文周、丁有和、曹弋、陈瀚、徐文胜、张为民、钱晓军、顾韵华、彭作民、高茜、陈冬霞、徐斌、王志瑞、孙德荣、周怡明、刘博宇、郑进、周何骏、陶卫冬、严大牛、刘启芬、邓拼搏、俞琰、周怡君、吴明祥、于金彬、马骏等。

由于作者水平有限，不当之处在所难免，恳请读者批评指正。

意见及建议邮箱：easybooks@163.com。

编 者

2015 年 6 月

目录

第 1 部分 Java 实用教程

第 1 章 Java 语言及编程环境	1	3.2.1 构造方法	49
1.1 Java 语言简介	1	3.2.2 默认构造方法	51
1.1.1 Java 语言特点	1	3.2.3 构造方法重载	52
1.1.2 Java 运行机制	2	3.3 this 引用	53
1.1.3 Java 8 新特性	3	3.4 静态成员	55
1.2 Java 编程环境	3	3.4.1 静态变量	55
1.2.1 JDK 8	3	3.4.2 静态方法	56
1.2.2 Eclipse 集成开发环境	7	3.4.3 main()方法	57
第 2 章 Java 语法基础	13	3.4.4 静态代码块	59
2.1 常量和变量	13	3.5 内部类	60
2.1.1 数据类型	13	3.5.1 实例内部类	60
2.1.2 标识符和关键字	13	3.5.2 匿名内部类	62
2.1.3 常量	14	3.5.3 static 内部类	63
2.1.4 变量	16	3.5.4 局部内部类	64
2.1.5 类型转换	17	3.6 类的打包封装	64
2.2 运算符和表达式	19	3.6.1 包(package)机制	64
2.2.1 运算符	19	3.6.2 创建 JAR 文件	68
2.2.2 表达式	24	第 4 章 Java 面向对象编程	70
2.3 流程控制	24	4.1 继承	70
2.3.1 分支语句	24	4.1.1 继承的定义	70
2.3.2 循环语句	27	4.1.2 初始化基类	71
2.3.3 流程跳转语句	30	4.1.3 方法的重写	72
2.4 数组	31	4.1.4 super 关键字	74
2.4.1 一维数组	31	4.2 多态	75
2.4.2 多维数组	33	4.2.1 对象的类型转换	75
2.4.3 数组的访问	36	4.2.2 多态的实现	78
2.5 综合实例	38	4.3 抽象类和接口	80
第 3 章 Java 类与对象	41	4.3.1 抽象方法与抽象类	80
3.1 Java 语言的类	41	4.3.2 接口概念及特性	81
3.1.1 面向对象程序设计概念	41	4.4 终止继承	84
3.1.2 类的定义	42	4.4.1 final 类	84
3.1.3 变量初始化	43	4.4.2 final 方法	84
3.1.4 类的方法	45	4.4.3 final 变量	85
3.2 创建对象	49	4.5 修饰符的适用范围	87

4.5.1	修饰符及访问权限	87	6.2.2	自定义注解	136
4.5.2	类内部	88	6.2.3	元注解	136
4.5.3	同一个包的类	89	6.3	lambda 表达式	139
4.5.4	不同包的子类	90	6.3.1	lambda 表达式简介	139
4.5.5	任意类	90	6.3.2	函数式接口	139
4.5.6	继承规则	91	6.3.3	lambda 表达式与匿名内 部类	141
4.6	Java 编程规范	91	6.3.4	方法引用	143
4.6.1	代码书写规范	91	第 7 章	容器和泛型	144
4.6.2	文档的使用	92	7.1	Collection 与 Iterator 接口	144
4.7	综合实例：航班管理	93	7.2	Collections 实用类	146
4.7.1	需求分析	93	7.3	Set (集合)	147
4.7.2	程序设计	94	7.3.1	HashSet	148
4.7.3	文档编制	103	7.3.2	TreeSet	149
第 5 章	Java 常用类	105	7.4	List (列表)	152
5.1	Object 类	105	7.4.1	ArrayList	153
5.1.1	equals()方法	105	7.4.2	LinkedList	155
5.1.2	hashCode()方法	106	7.4.3	栈的实现	156
5.1.3	toString()方法	107	7.5	Queue (队列)	157
5.2	字符串类	108	7.5.1	LinkedList 实现	157
5.2.1	String 类	108	7.5.2	PriorityQueue	158
5.2.2	StringBuffer 类	111	7.5.3	双向队列	159
5.3	包装类	112	7.6	Map (映射)	161
5.4	Math 类	114	7.6.1	HashMap	161
5.5	Random 类	115	7.6.2	TreeMap	162
5.6	日期时间类	116	7.7	泛型	163
5.6.1	Java 8 的日期时间包	116	7.7.1	泛型的基本概念	163
5.6.2	日历应用	118	7.7.2	泛型方法	166
5.7	正则表达式	119	7.7.3	受限泛型	168
5.7.1	基础知识	119	7.7.4	通配符与受限通配符	169
5.7.2	正则表达式的应用	122	7.7.5	子类泛型	170
5.8	数组实用类	125	7.7.6	泛型数组	172
5.8.1	复制数组	125	7.8	综合实例	173
5.8.2	数组排序	126	第 8 章	异常处理	177
5.8.3	数组元素的查找	127	8.1	异常概述	177
第 6 章	Java 语言新特性	129	8.2	异常处理机制	178
6.1	枚举	129	8.2.1	异常的捕获与处理	178
6.1.1	定义枚举类型	129	8.2.2	声明抛出异常子句	180
6.1.2	enum 构造方法	130	8.2.3	抛出异常	180
6.1.3	使用 EnumMap	131	8.3	自定义异常类	181
6.2	注解	132	第 9 章	Java 输入/输出系统	184
6.2.1	内置注解	132			

9.1 字节流	184	11.1.1 Window 和 Frame	235
9.1.1 字节数组输入流	185	11.1.2 Panel	235
9.1.2 字节数组输出流	186	11.2 布局管理器	237
9.1.3 文件输入流	186	11.3 事件处理机制	239
9.1.4 文件输出流	187	11.3.1 AWT 事件与监听器	239
9.1.5 管道流	188	11.3.2 窗口事件	240
9.2 过滤流	189	11.3.3 鼠标事件	242
9.2.1 缓冲流类	190	11.3.4 键盘事件	243
9.2.2 数据流类	191	11.3.5 内部类实现监听接口	245
9.2.3 PrintStream 类	193	11.3.6 类自身实现监听接口	246
9.3 字符流	194	11.3.7 外部类实现监听接口	247
9.3.1 转换流类	195	11.3.8 采用事件适配器	249
9.3.2 FileReader 和 FileWriter	196	11.4 在 AWT 中绘图	251
9.3.3 BufferedReader 和 BufferedWriter	197	11.4.1 Graphics 类	251
9.3.4 PrintWriter 类	198	11.4.2 AWT 绘图实现原理	253
9.4 标准 I/O	200	11.5 综合实例	255
9.4.1 重新包装标准 I/O	200	第 12 章 Swing 组件及应用	260
9.4.2 标准 I/O 重定向	200	12.1 窗口 (JFrame)	260
9.5 随机访问文件类	201	12.2 Swing 基本组件	261
9.6 对象序列化	204	12.2.1 按钮 (JButton)	261
9.7 File 类	205	12.2.2 文本框 (JTextField)	263
9.8 综合实例	207	12.2.3 复选框 (JCheckBox) 和单 选按钮 (JRadioButton)	266
第 10 章 多线程	209	12.2.4 列表框 (JList)	269
10.1 线程的创建与启动	209	12.3 Swing 菜单程序设计	270
10.1.1 继承 java.lang.Thread 类	209	12.3.1 菜单条 (JMenuBar)	270
10.1.2 实现 Runnable 接口	211	12.3.2 弹出式菜单 (JPopupMenu)	273
10.2 线程的状态转换	212	12.3.3 页签面板 (JTabbedPane)	275
10.3 线程调度	214	12.3.4 工具栏 (JToolBar)	277
10.3.1 调整线程优先级	214	12.4 Swing 数据管理组件	279
10.3.2 线程让步	215	12.4.1 表格 (JTable)	280
10.3.3 合并线程	216	12.4.2 树状视图 (JTree)	282
10.4 后台线程	217	12.5 Swing 标准对话框	283
10.5 线程互斥	218	12.5.1 对话框 (JDialog)	283
10.5.1 临界区	218	12.5.2 消息框 (JOptionPane)	285
10.5.2 对象锁机制	220	12.5.3 文件对话框 (JFileChooser)	287
10.5.3 显式 Lock 对象	223	12.6 综合实例	289
10.6 线程同步	225	第 13 章 Java 基础开发综合实习	293
10.7 综合实例	228	13.1 设计思路	293
第 11 章 AWT 图形用户界面编程	234	13.2 汉诺塔上盘子模拟	293
11.1 AWT 容器	234		

13.3	对象定位及盘子的存放	294	14.4.1	URL 类	322
13.4	创建汉诺塔及实现手动搬运盘子	295	14.4.2	URLConnection 类	324
13.5	自动搬运盘子	301	14.5	综合实例	326
第 14 章	Java 网络编程	307	第 15 章	Java 数据库编程	331
14.1	网络程序设计基础	307	15.1	SQL 语言简介	331
14.1.1	TCP 与 UDP	307	15.2	JDBC 原理	332
14.1.2	端口和套接字	307	15.3	访问数据库	333
14.2	TCP 网络编程	308	15.3.1	加载并注册数据库驱动	334
14.2.1	TCP 通信模型	308	15.3.2	建立到数据库的连接	334
14.2.2	Socket 编程相关类	308	15.3.3	执行数据库操作	335
14.2.3	TCP 通信程序	311	15.4	JDBC 编程	337
14.3	UDP 网络编程	314	15.4.1	创建编程环境	337
14.3.1	UDP 相关类	314	15.4.2	创建数据库和表	337
14.3.2	UDP 通信程序	316	15.4.3	操作数据库	339
14.3.3	组播	319	15.4.4	事务处理	341
14.4	URL 网络编程	322	15.5	综合实例	343

第 2 部分 实验

实验 1	Java 语言及编程环境	348	实验 8	Java 异常处理	365
实验 2	Java 语法基础	349	实验 9	Java 输入/输出系统	366
实验 3	Java 类与对象	353	实验 10	多线程	370
实验 4	Java 面向对象编程	356	实验 11	AWT 图形用户界面编程	373
实验 5	Java 常用类	358	实验 12	Swing 组件及应用	378
实验 6	Java 语言新特性	360	实验 13	Java 网络编程	384
实验 7	容器和泛型	362	实验 14	Java 数据库编程	390

第 3 部分 习题集

第 1 章	Java 语言及编程环境	393	第 8 章	异常处理	395
第 2 章	Java 语法基础	393	第 9 章	Java 输入/输出系统	395
第 3 章	Java 类与对象	394	第 10 章	多线程	395
第 4 章	Java 面向对象编程	394	第 11 章	AWT 图形用户界面编程	396
第 5 章	Java 常用类	394	第 12 章	Swing 组件及应用	397
第 6 章	Java 语言新特性	394	第 14 章	Java 网络编程	397
第 7 章	容器和泛型	395	第 15 章	Java 数据库编程	397

第1部分 Java 实用教程

第1章 Java 语言及编程环境

1.1 Java 语言简介

Java 是由原 Sun 公司（现已被甲骨文公司收购）于 1991 年开发的编程语言，初衷是为智能家电的程序设计提供一个分布式代码系统。为了使整个系统与平台无关，采用了虚拟机器码方式，虚拟机内运行解释器，而针对每种操作系统均有其对应的解释器，这样 Java 就成了与平台无关的语言。后来 Java 技术被广泛应用于万维网，伴随因特网的普及迅速成长起来，取得了惊人的发展。1995 年 5 月 23 日，Java 语言诞生，次年 1 月发布了第一个开发包 JDK 1.0，至 JDK 1.1（1997.2.18）奠定了 Java 在计算机语言中的地位。从 JDK 1.2（1998.12.8）开始，Java 的应用平台逐步分化，演进成 3 个版本。

- **Java SE**—Java Standard Edition（标准版），主要用于普通 PC 机、工作站的 Java 控制台或桌面程序的基础开发。
- **Java ME**—Java Micro Edition（微型版），用于移动设备、嵌入式设备上的 Java 应用程序开发。
- **Java EE**—Java Enterprise Edition（企业版），用于开发、部署和管理企业级、可扩展的大型软件或 Web 应用。

本书程序设计涉及的只是 Java SE。2004 年 9 月 30 日发布的 Java SE 1.5（基于 JDK 1.5）对 Java 语言本身进行了很大革新，引入许多新的概念和特性，成为 Java 发展史上的里程碑。为显示这个版本的重要，Sun 将其更名为 Java SE 5.0。在 2005 年 6 月召开的 JavaOne 大会上，Sun 公司又公开了 Java SE 6.0。2009 年 4 月 20 日，甲骨文公司以 74 亿美元天价收购了 Sun 公司并取得了 Java 的版权，之后又持续推出 Java SE 7.0（2011.7.28）和 Java SE 8.0（2014.3.18），Java 技术日臻成熟完善。如今 Java 已然成为计算机软件开发领域最流行的语言和平台之一。本教材就以 **Java SE 8.0**（基于最新 JDK 1.8.0，简称 Java 8）为编程环境，全面系统地来讲授 Java 程序设计的知识。

1.1.1 Java 语言特点

Java 是一个广泛使用的网络编程语言，它简单、面向对象，不依赖于机器结构，不受 CPU 和环境限制，具有可移植性、安全性，并提供了多线程机制，具有很高的性能。此外，Java 还提供了丰富的类库，使程序设计人员能很方便地建立自己的系统。概括起来说，Java 语言具有如下特点。

(1) **简单性**。Java 语言虽衍生自 C/C++, 但它略去了 C/C++ 中指针、运算符重载、多重继承等复杂的概念, 并通过自动垃圾收集机制大大简化了程序员的内存管理工作。另外, Java 对环境的要求非常低, 它的基本解释器及类仅几十个 KB, 加上标准类库和线程的支持也只有大约 200KB。

(2) **面向对象**。Java 是一个完全面向对象的语言, 其程序设计集中于对象及其接口, 它提供了简单的类机制及动态的接口模型。对象中封装了它的状态变量及相应方法, 实现了模块化和信息隐藏; 而类则提供了一类对象的原型, 通过继承机制, 子类可使用父类的方法, 实现代码复用。

(3) **分布式**。Java 是面向网络的语言, 通过它提供的类库可以处理 TCP/IP 协议, 用户可以通过 URL 地址在网络上很方便地访问其他对象。

(4) **安全性**。因 Java 不支持指针, 一切对内存的访问都必须通过对象的实例来完成, 这就有效防止了黑客使用“特洛伊”木马等欺骗手段访问对象的私有成员, 同时也避免了由于指针操作失误导致的程序或系统崩溃。

(5) **可移植性**。与平台无关的特性使 Java 程序可以很方便地被移植到不同硬件平台的计算机上。同时, Java 自身的类库也实现了与不同平台的接口, 使这些类库更容易移植。

(6) **高性能**。和其他解释执行的语言不同, Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码, 从而得到较高的性能。

(7) **支持多线程**。多线程机制使应用程序能够并发地执行, 同步机制又保证了对共享数据的正确操作。通过使用多线程, 程序员可以分别用不同的线程完成特定的行为, 而不需要采用全局的事件循环机制, 这样就很容易地实现网络上的实时交互行为。

1.1.2 Java 运行机制

Java 有两个核心的运行机制: 一个是 Java 虚拟机 (Java Virtual Machine, JVM); 另一个是垃圾收集机制 (Garbage Collection)。

1. Java 虚拟机

Java 虚拟机 (JVM) 可理解成一个以字节码为机器指令的 CPU。首先, Java 编译程序将后缀名为 .java 的 Java 源程序编译为 JVM 可执行的代码 (后缀名为 .class 的 Java 字节码文件), 如图 1.1 所示, 运行 JVM 字节码的工作则由解释器来完成。整个运行过程分代码的装入、校验和执行三步进行: 装入代码的工作由类装载器完成, 类装载器负责装入一个程序运行所需要的所有代码; 字节码校验器负责代码的校验; 解释器负责代码的执行。每种类型的操作系统都有一种对应的 JVM, JVM 屏蔽了底层操作系统的差异, 使 Java 程序能够做到“一次编译, 到处运行”。

2. 垃圾收集器

Java 垃圾收集器能够自动回收垃圾, 即运行时无用对象占据的内存空间。而在 C/C++ 中, 垃圾收集工作全都要由程序员负责, 这无疑增加了程序员的负担。Java 语言通过提供一种系统级线程来自动跟踪程序运行时存储空间的分配情况, 并在 JVM 空闲时检查并释放那些可被释放的存储空间。在 Java 中, 对象一旦被创建就会在堆区中分配一块内存, 而当对象不再被程序引用时, 它就变成一个“垃圾”, 所占用的堆空间可被回收以便腾出来给后续的新对象使用。Java 垃圾收集器能断定哪些对象不再被引用, 并且能够把它们所占据的堆空间释放出来。

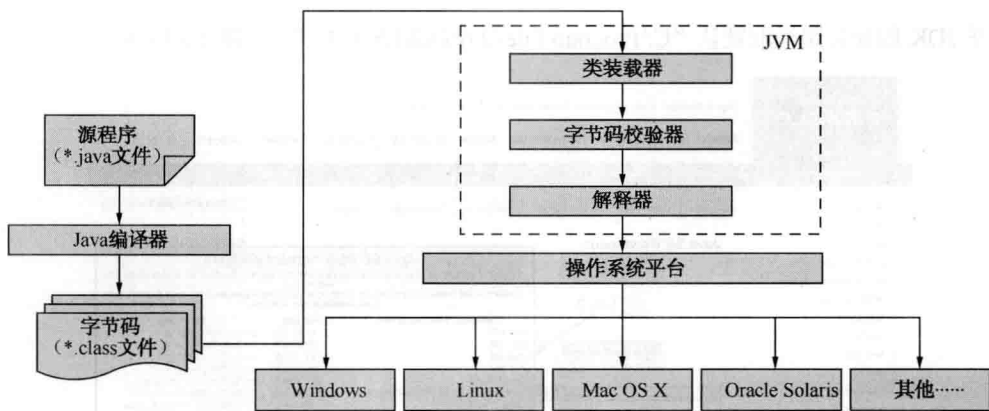


图 1.1 Java 程序执行机制

1.1.3 Java 8 新特性

Java 8 是继 Java SE 5.0 以来对 Java 语言的又一次重要升级，包含了许多令人欣喜的新特性，下面简单列出主要的几个。

(1) **lambda 表达式**。这种新的语法为 Java 添加了函数式编程特性，可以简化并减少程序中创建特定结构（如某些类型的匿名类）所需的代码量，使得 Java 语言更为灵活和富有生命力。

(2) **新的流 API**。流 API 支持对数据执行管道操作，并针对 lambda 表达式做了优化。

(3) **简化的接口实现**。Java 8 中可以为接口指定的方法定义默认实现，如果程序员没有为默认方法创建实现，就使用接口定义的默认实现，在向接口添加新方法时也不会破坏现有方法。

(4) **新的时间日期 API**。新增日期时间格式器增强了对时间日期类型数据的处理功能。

(5) **支持 JavaFX**。JavaFX 是 Java 新一代的 GUI 框架，它强大灵活，可制作出视觉效果十分出色的应用程序。此次升级的 Java 8 捆绑了对 JavaFX 8 的支持，以适应未来 Java 图形界面开发由 AWT/Swing 向 JavaFX 的平稳过渡。

Java 8 新增的功能还有很多，本教材将在后续章节结合有关知识点逐一加以介绍。

1.2 Java 编程环境

1.2.1 JDK 8

要想编译和运行 Java 程序，离不开 Java 的编译和运行环境。Sun 公司（今为甲骨文公司）为 Java 提供一套原生的开发环境，通常称为 JDK（Java SE Development Kits, JSDK）。目前最新的版本是 Java SE Development Kit 8 Update 45，即 JDK 8。

1. 下载 JDK 8

可以到甲骨文公司的官网下载 JDK8：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>，单击“DOWNLOAD”按钮，如图 1.2 所示，在出现的下载页上点击“jdk-8u45-windows-i586.exe”链接，下载对应 Windows x86 体系计算机的 JDK。

2. 安装 JDK 8

下载完成后，得到可执行文件 jdk-8u45-windows-i586.exe，双击启动安装向导，按照提示完成安

装，这里 JDK 的安装路径取默认“C:\Program Files\Java\jdk1.8.0_45\”，如图 1.3 所示。



图 1.2 选择下载 JDK

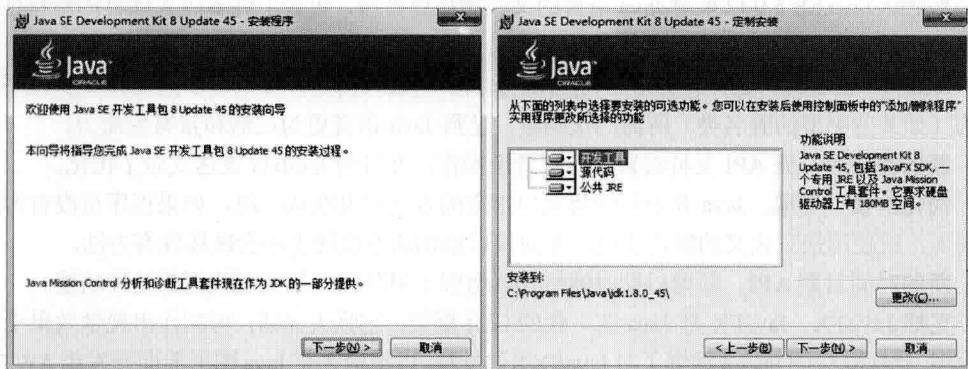


图 1.3 安装 JDK

在 JDK 的安装包里带有 JRE 安装包，JRE 是 Java 运行时的环境，默认安装到“C:\Program Files\Java\jre1.8.0_45\”。

3. 配置环境变量

JDK 安装完成之后，还需要配置环境变量才可使用，下面是具体设置步骤。

(1) 打开“环境变量”对话框

右击桌面“计算机”图标，选择“属性”命令，在弹出的控制面板主页中点击“高级系统设置”链接，在弹出的“系统属性”对话框里单击“环境变量”按钮，弹出“环境变量”对话框，操作如图 1.4 所示。

(2) 新建系统变量 JAVA_HOME

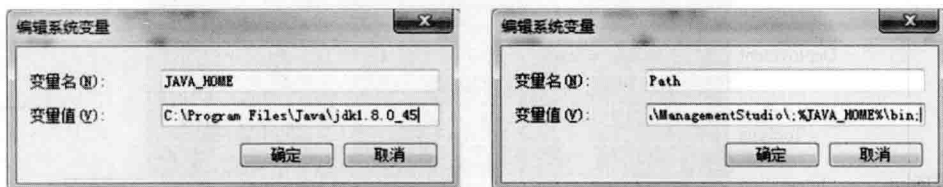
在“系统变量”列表下单击“新建”按钮，弹出“新建系统变量”对话框。在“变量名”一栏输入“JAVA_HOME”，“变量值”栏输入 JDK 安装路径，如图 1.5 (a) 所示，单击“确定”按钮。

(3) 设置系统变量 Path

在“系统变量”列表中找到名为“Path”的变量，单击“编辑”按钮，弹出“编辑系统变量”对话框，在“变量值”字符串中加入路径“%JAVA_HOME%\bin;”，如图 1.5 (b) 所示，单击“确定”按钮。



图 1.4 打开“环境变量”对话框



(a) 新建 JAVA_HOME 变量

(b) 编辑 Path 变量

图 1.5 设置环境变量

选择任务栏“开始”→“运行”命令，输入“cmd”回车，在命令行输入“java -version”回车，如果环境变量设置成功就会出现 Java 的版本信息，如图 1.6 所示。

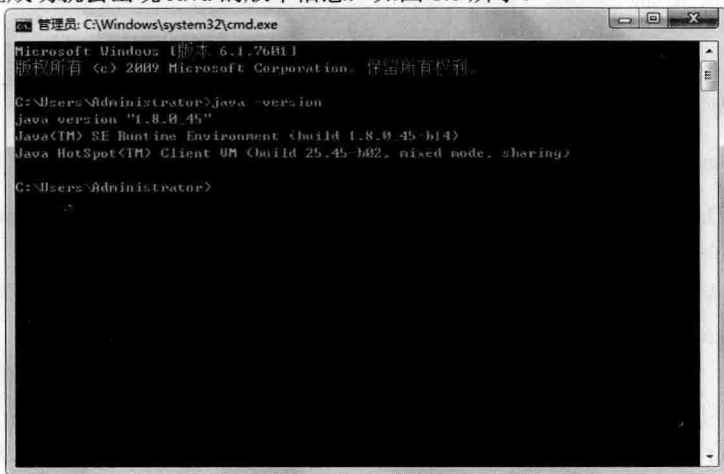


图 1.6 JDK 8 配置成功

4. JDK 8 的构成

进入 JDK 8 的安装路径，将看到如表 1.1 所示 JDK8 的目录结构。

表 1.1 JDK 8 的目录结构

目 录	描 述
bin 子目录	存放 JDK 8 的工具程序
db 子目录	存放 Apache Derby 数据库等开放资源，支持 JDBC4.0 规范
include 子目录	存放与 C 程序相关的头文件
jre 子目录	存放 Java 运行时环境（JRE）相关的文件
lib 子目录	存放 Java 类库（JAR 文件）

JDK 8 包含的内容非常丰富，图 1.7 是其官方文档所提供的概念体系结构图，从中可一窥 JDK 8 的构成细节。从图上看，Java 实质是一系列工具、庞大的类库及 JVM 集成在一起组成的复杂编程环境，但概括起来，有以下几大基本构成。

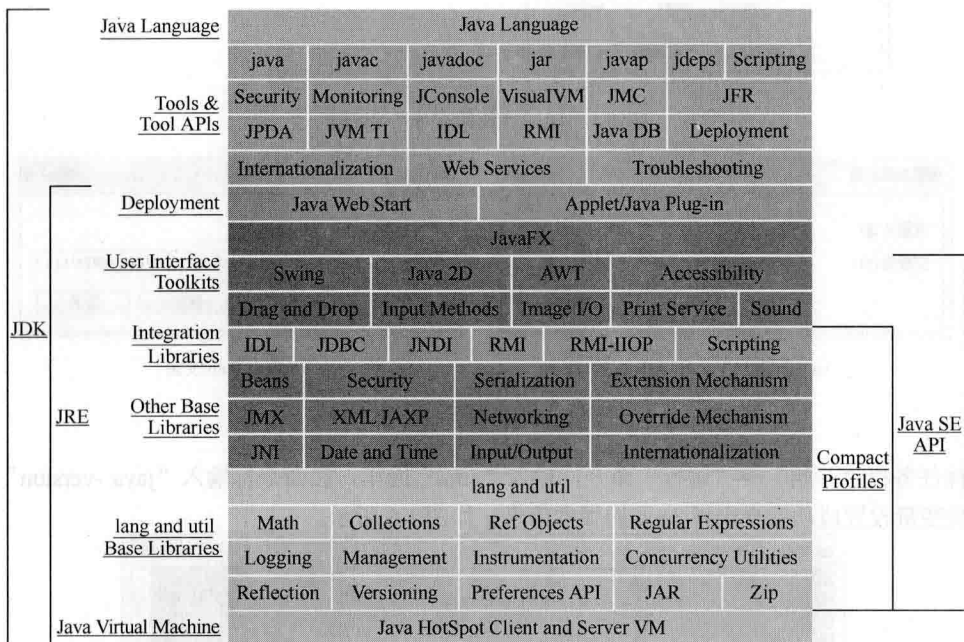


图 1.7 Java Platform Standard Edition 8 概念体系结构图（Copyright © 1993-2015, Oracle）

（1）JVM：即 Java 虚拟机，它位于图 1.7 整个体系的底层，负责解释、执行 Java 程序，可以运行在各种操作系统平台上。

（2）JDK 8 类库：JVM 之上是 Java 最基础的类库，提供了各种实用类，包括最常用的 java.lang、java.util、java.io、java.sql 和 javax.swing 等。

（3）开发工具：在图 1.7 体系的上层是种类繁多的开发工具，这些工具随 Java 官方发布的 JDK 一起打包提供，都是可执行的程序，并公开了 APIs（使用命令接口），常用的工具有 javac.exe（编译工具）、java.exe（运行工具）、javadoc.exe（用于生成 Javadoc 文档）和 jar.exe（打包工具）等。

1.2.2 Eclipse 集成开发环境

事实上, 仅用记事本及 JDK 自带的工具 (javac.exe、java.exe 等) 就足以编写 Java 程序和进行 Java 软件开发, 但为提高效率, 通常不这么做, 而是借助于现成的、功能更为强大的集成开发环境 (IDE)。目前比较主流的 IDE 有 Eclipse、MyEclipse 和 NetBeans 等, 本书选用免费开源的 Eclipse 作为 Java 语言的开发环境。

1. 安装 Eclipse 4.4

可从 Eclipse 官网下载: <http://www.eclipse.org/downloads/>, 当前最新的发布版本是 Eclipse 4.4.2, 下载后直接解压即可使用。解压后, 在磁盘上生成一个 eclipse 文件夹, 进入双击 eclipse.exe, 出现如图 1.8 所示的欢迎界面。

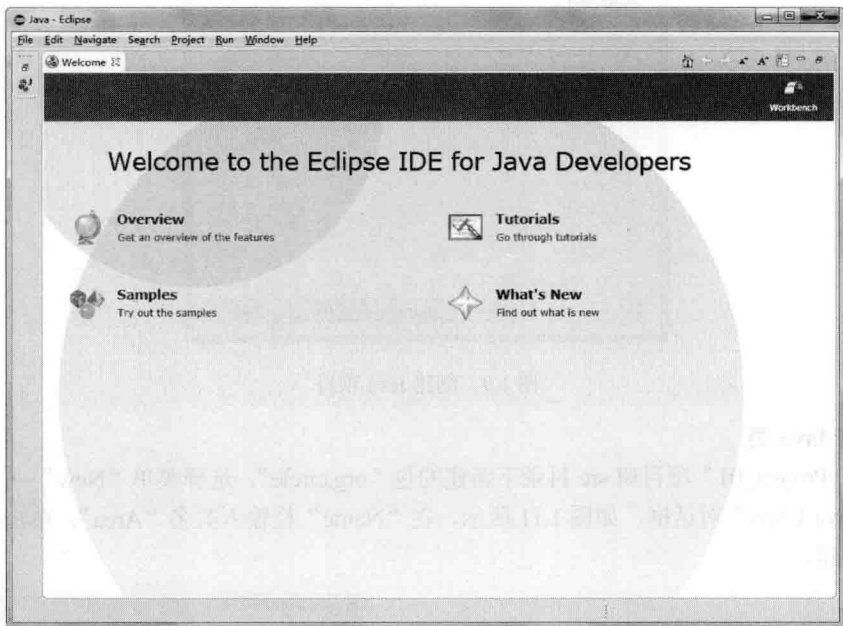


图 1.8 Eclipse 欢迎界面

为使用方便, 右击 eclipse.exe, 选择“发送到”→“桌面快捷方式”命令, 在桌面上将出现 Eclipse 的快捷方式, 以后直接双击快捷方式即可启动 Eclipse。

2. 第一个 Java 程序

现在就用 Eclipse 来编写第一个 Java 程序, 借此熟悉它的使用。Eclipse 以“项目 (Project)”的形式管理 Java 程序, 开发一个基本的 Java 项目包括三个步骤。

(1) 创建 Java 项目

启动 Eclipse, 在其工作窗口中依次选择主菜单“File”→“New”→“Java Project”项, 打开“New Java Project”向导, 如图 1.9 所示, 在“Project name”栏输入项目名“MyProject_01”, 其他选项默认, 单击“Finish”按钮, 项目创建成功。项目“MyProject_01”将出现在左边的“Package Explorer” (包资源管理器) 中。

(2) 创建 Java 包

在“包资源管理器”中右击新创建的项目“MyProject_01”, 选择菜单“New”→“Package”项,

弹出“New Java Package”对话框,如图 1.10 所示,在“Name”栏输入包名“org.circle”,单击“Finish”按钮完成包的创建。

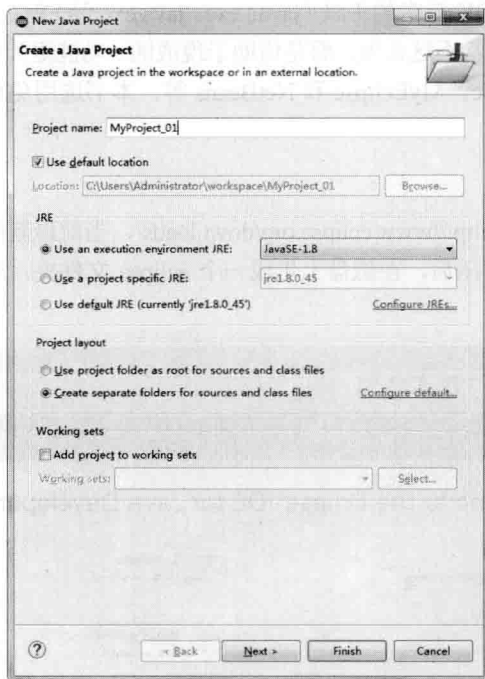


图 1.9 创建 Java 项目

(3) 创建 Java 类

右击“MyProject_01”项目树 src 目录下新建的包“org.circle”,选择菜单“New”→“Class”项,弹出“New Java Class”对话框,如图 1.11 所示,在“Name”栏输入类名“Area”,单击“Finish”按钮完成类的创建。

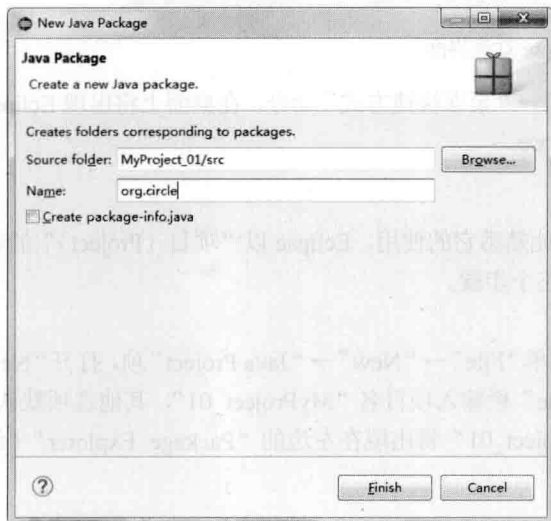


图 1.10 创建 Java 包



图 1.11 创建 Java 类

系统自动在中央工作区打开 Area 类的代码编辑窗口, 这时就可以编写 Java 程序了。输入“Area.java”源程序, 如图 1.12 所示, 完成后单击工具栏“Save”() 按钮保存。



图 1.12 编写 Java 代码

右击“Area.java”, 选择菜单“Run As”→“Java Application”项, 运行 Java 程序, 在下方控制台区显示输出结果:

圆的面积=28.2744

3. 程序分析

下面来看这个 Java 小程序的代码, 简要分析一下, 使大家对 Java 编程有一个初步的认识。

【例 1.1】求圆面积。

源程序文件名为 Area.java, 代码如下。

Area.java

```
package org.circle;
/**                                     // (1)
 * Title: 求圆的面积
 * Description: 已知圆的半径 r, 求圆的面积
 * Copyright: Copyright (c) 2015
 * Company: 南京师范大学
 * @author 郑阿奇
 * @version 3.0
 */
public class Area {                     // (2)
```