

高等学校**计算机**教材

· 含视频教学 ·

Java

实用教程 (第4版)

郑阿奇 / 主编

姜乃松 殷红先 刘颖 张卫芬 / 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4版)(含视频教学)

郑阿奇 主编

姜乃松 殷红先 刘颖 张卫芬 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以最新长期支持稳定版本 Java 为基础,对第 3 版进行了升级;除保留了原版内容的循序渐进、深入浅出等诸多优点外,精心设计每个实例,囊括了 Java 的最新语言特性;增强了 Java 数据库编程部分的内容,涉及操作最新 MySQL、SQL Server、Oracle 等多种最新主流数据库的关键技术。

本书由 Java 实用教程、实验、习题集这三个部分组成。第 1 部分:一般在介绍内容后紧跟相关的实例演示,在第 2、4、7、9~12、14、15 章章尾配有综合实例,以提高读者综合应用已学过的主要知识的能力。第 2 部分:首先通过实例引导读者学习,然后提出思考问题,最后在原有基础上让读者自己进行操作和编程练习。第 3 部分:本版系统地扩充了习题的内容,对应于主要知识点,对所有习题均结合书中重要知识点进行了精心设计;通过扫描书中的二维码可得到配套答案,帮助读者在巩固所学知识的同时提高能力。

本书以最新版 Eclipse 作为 Java 的集成开发环境,该版本与各版 JDK 的兼容性较好,能将用户以旧版 JDK 开发的 Java 项目自动转换成新版项目,其编写、调试、运行 Java 程序都十分方便。本书所有实例均在最新版 Eclipse 下调试通过,实例运行结果以抓图形式提供。

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。本书对每个重要知识点均提供通过二维码播放的配套视频教学,既可作为课后教学辅助,也可作为课程自学。

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:含视频教学 / 郑阿奇主编. —4 版. —北京:电子工业出版社,2019.12
ISBN 978-7-121-37945-1

I. ①J… II. ①郑… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 254982 号

责任编辑:程超群 特约编辑:张燕虹

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

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:26 字数:699 千字

版 次:2005 年 4 月第 1 版

2019 年 12 月第 4 版

印 次:2019 年 12 月第 1 次印刷

定 价:79.00 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, 88258888。

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

本书咨询联系方式:(010) 88254577, 91034@qq.com。

前 言

Java 是计算机领域中最主流的编程语言之一，其简单、面向对象、多线程、跨平台等特性深受人们的喜爱。Java 语言拥有全球最为庞大的开发者队伍，在互联网开发、移动开发、大数据、人工智能等众多领域中都具有广泛的应用。

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。2009 年，修订、出版了第 2 版。2015 年，根据当时发布的 Java 8 对上一版进行全面修改后出版了第 3 版。如今，因 Java 最新稳定版为 Java 11，它具有很多新特性，故需要对上一版进行及时更新。

本书以甲骨文公司发布的最新长期支持（Long Term Support, LTS）稳定版本 Java 11 为基础，对第 3 版进行了升级；除保留原版内容的循序渐进、深入浅出等诸多优点外，还精心设计每个实例，加入了 Java 的最新语言特性；增强了 Java 数据库编程部分的内容，涉及操作 MySQL、SQL Server、Oracle 等多种最新主流数据库的关键技术，也可供从事 Java 跨平台异构系统开发的读者参考。

本书由 Java 实用教程、实验、习题集这三个部分组成。第 1 部分：一般在介绍内容后紧跟相关的实例演示，在第 2、4、7、9~12、14、15 章章尾配有综合实例，以提高读者综合应用已学过的主要知识的能力。第 2 部分：首先通过实例引导读者学习，然后提出思考问题，最后在原有基础上让读者自己进行操作和编程练习。第 3 部分：本版系统地扩充了习题的内容，对应于主要知识点，对所有习题均结合书中重要知识点进行了精心设计；通过扫描书中的二维码可得到配套的答案，帮助读者在巩固所学的知识的同时提高能力。

本书以最新版 Eclipse 作为 Java 的集成开发环境，该版本与各版 JDK 的兼容性较好，能将用户以旧版 JDK 开发的 Java 项目自动转换成新版项目，编写、调试、运行 Java 程序都十分方便。本书所有实例均在最新版 Eclipse 下调试通过，实例运行结果以抓图形式提供。

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 自学者或应用开发者的参考书。

实际上，本书不仅适合教学，也非常适合 Java 的各类培训，以及供使用 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。书中的源代码工程和电子课件，可从华信教育资源网（<http://www.hxedu.com.cn>）上免费下载。本书对每个重要知识点均提供通过二维码播放的配套视频教学，既可以作为课后教学辅助，也可作为课程自学。

本书由南京师范大学的郑阿奇担任主编，由姜乃松、殷红先、刘颖和张卫芬（东南大学成贤学院）编著。参加编写的还有周何骏、孙德荣、王钢花、刘美芳、郑博琳、刘忠等人，在此表示诚挚的谢意。

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 著 者

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为，歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

本书视频目录

(建议在 WiFi 环境下扫码观看)

序号	文件 名	序号	文件 名
1	(2.1) 常量和变量.mp4	33	(6.3) lambda 表达式.mp4
2	(2.2) 运算符和表达式.mp4	34	(7.4) List (列表) .mp4
3	(2.3) 流程控制_条件语句程序设计思维过程展现.mp4	35	(7.6) Map (映射) .mp4
4	(2.3) 流程控制_switch 语句程序设计思维过程展现.mp4	36	(7.9) 综合实例.mp4
5	(2.3) 流程控制_while 语句程序设计思维过程展现.mp4	37	(8.2) 异常处理机制.mp4
6	(2.3) 流程控制_do-while 语句程序设计思维过程展现.mp4	38	(9.1) 字节流_内存流的解析与调序设计示例.mp4
7	(2.3) 流程控制_for 语句程序设计思维过程展现.mp4	39	(9.1) 字节流_文件流的解析与设计示例.mp4
8	(2.4) 数组_一维数组解析及程序设计思维过程展现.mp4	40	(9.2) 过滤流_缓冲流解析及程序设计示例.mp4
9	(2.4) 多维数组_多维数组解析及程序设计思维过程展现_1.mp4	41	(9.2) 过滤流_Java 基本数据类型流程序设计示例.mp4
10	(2.4) 多维数组_多维数组解析及程序设计思维过程展现_2.mp4	42	(9.3) 字符流_转换流解析及示例.mp4
11	(2.5) 综合实例_程序设计思维能力训练.mp4	43	(9.3) 字符流_文本文件流解析与示例.mp4
12	(3.1) Java 语言的类_类的定义.mp4	44	(9.5) 随机访问文件类.mp4
13	(3.1) Java 语言的类_类的方法_递归解析及能力训练 1.mp4	45	(9.6) 对象序列化.mp4
14	(3.1) Java 语言的类_类的方法_递归解析及能力训练 2.mp4	46	(9.7) File 类及程序设计示例.mp4
15	(3.1) Java 语言的类_方法重载.mp4	47	(9.8) Java 输入输出综合程序设计能力训练.mp4
16	(3.2) 创建对象_(3.3) this 引用.mp4	48	(10.1) 线程的创建与启动.mp4
17	(3.4) 静态成员.mp4	49	(10.5) 线程互斥.mp4
18	(3.5) 内部类.mp4	50	(10.6) 线程同步.mp4
19	(3.6) 类的打包封装.mp4	51	(11.2) 布局管理器.mp4
20	(4.1) 继承.mp4	52	(11.3) 事件处理机制.mp4
21	(4.2) 多态.mp4	53	(11.4) 在 AWT 中绘图.mp4
22	(4.3) 抽象类和接口.mp4	54	(12.1) 窗口 (JFrame) .mp4
23	(4.5) 修饰符的适用范围.mp4	55	(12.2) Swing 基本组件之一.mp4
24	(4.6) Java 编程规范.mp4	56	(12.2) Swing 基本组件之二.mp4
25	(4.7) 综合实例: 航班管理.mp4	57	(12.3) Swing 菜单程序设计.mp4
26	(5.2) 字符串类.mp4	58	(12.5) Swing 标准对话框.mp4
27	(5.3) 包装类.mp4	59	(13.2) Java 基础开发综合实习_Swing 综合程序设计训练——汉诺塔盘子模拟.mp4
28	(5.4) Math 类_(5.5) Random 类.mp4	60	(14.2) TCP 网络编程.mp4
29	(5.6) 日期时间类.mp4	61	(14.3) UDP 网络编程.mp4
30	(5.7) 正则表达式_基本正规式的表示及程序设计训练.mp4	62	(14.4) URL 网络编程.mp4
31	(5.7) 正则表达式_Pattern、Matcher 及程序设计能力训练.mp4	63	(15.4) JDBC 程序设计与示例.mp4
32	(6.1) 枚举.mp4		

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 创建对象	55
1.1 Java 语言简介	1	3.2.1 构造方法	55
1.1.1 Java 语言特点	1	3.2.2 默认构造方法	57
1.1.2 Java 运行机制	2	3.2.3 构造方法重载	58
1.1.3 从 Java 5 到 Java 11	3	3.3 this 引用	59
1.2 Java 编程环境	4	3.4 静态成员	61
1.2.1 JDK 11	4	3.4.1 静态变量	61
1.2.2 Eclipse 集成开发环境	6	3.4.2 静态方法	62
第 2 章 Java 语法基础	19	3.4.3 main()方法	63
2.1 常量和变量	19	3.4.4 静态代码块	65
2.1.1 数据类型	19	3.5 内部类	65
2.1.2 标识符和关键字	19	3.5.1 实例内部类	66
2.1.3 常量	20	3.5.2 匿名内部类	68
2.1.4 变量	22	3.5.3 static 内部类	69
2.1.5 类型转换	23	3.5.4 局部内部类	70
2.2 运算符和表达式	25	3.6 类的打包或封装	70
2.2.1 运算符	25	3.6.1 包(package)机制	70
2.2.2 表达式	30	3.6.2 创建 JAR 文件	74
2.3 流程控制	30	第 4 章 Java 面向对象编程	76
2.3.1 分支语句	30	4.1 继承	76
2.3.2 循环语句	33	4.1.1 继承的定义	76
2.3.3 流程跳转语句	36	4.1.2 初始化基类	77
2.4 数组	37	4.1.3 方法的重写	78
2.4.1 一维数组	37	4.1.4 super 关键字	80
2.4.2 多维数组	39	4.2 多态	81
2.4.3 数组的访问	42	4.2.1 对象的类型转换	81
2.5 综合实例	44	4.2.2 多态的实现	84
第 3 章 Java 类与对象	47	4.3 抽象类和接口	86
3.1 Java 语言的类	47	4.3.1 抽象方法与抽象类	86
3.1.1 面向对象程序设计概念	47	4.3.2 接口概念及特性	87
3.1.2 类的定义	48	4.4 终止继承	89
3.1.3 变量初始化	49	4.4.1 final 类	90
3.1.4 类的方法	51	4.4.2 final 方法	90

4.4.3	final 变量	91	6.2	注解	138
4.5	修饰符的适用范围	93	6.2.1	内置注解	138
4.5.1	修饰符及访问权限	93	6.2.2	自定义注解	142
4.5.2	类内部	93	6.2.3	元注解	142
4.5.3	同一个包的类	94	6.3	lambda 表达式	145
4.5.4	不同包的子类	95	6.3.1	lambda 表达式简介	145
4.5.5	任意类	96	6.3.2	函数式接口	146
4.5.6	继承规则	97	6.3.3	lambda 表达式与匿名 内部类	147
4.6	Java 编程规范	97	6.3.4	方法引用	149
4.6.1	代码书写规范	97	第 7 章	容器和泛型	150
4.6.2	文档的使用	98	7.1	Collection 与 Iterator 接口	150
4.7	综合实例: 航班管理	99	7.2	Collections 实用类	152
4.7.1	需求分析	99	7.3	Set (集合)	153
4.7.2	程序设计	100	7.3.1	HashSet 类	153
4.7.3	文档编制	109	7.3.2	TreeSet 类	155
第 5 章	Java 常用类	111	7.4	List (列表)	158
5.1	Object 类	111	7.4.1	ArrayList 类	158
5.1.1	equals()方法	111	7.4.2	LinkedList 类	161
5.1.2	hashCode()方法	112	7.4.3	栈的实现	162
5.1.3	toString()方法	113	7.5	Queue (队列)	163
5.2	字符串类	114	7.5.1	LinkedList 类实现	163
5.2.1	String 类	114	7.5.2	PriorityQueue 类	163
5.2.2	StringBuffer 类	118	7.5.3	双向队列	165
5.3	包装类	119	7.6	Map (映射)	166
5.4	Math 类	121	7.6.1	HashMap	166
5.5	Random 类	122	7.6.2	TreeMap	167
5.6	日期时间类	123	7.7	泛型	169
5.6.1	新日期时间包	123	7.7.1	泛型的基本概念	169
5.6.2	日历应用	124	7.7.2	泛型方法	171
5.7	正则表达式	125	7.7.3	受限泛型	173
5.7.1	基础知识	125	7.7.4	通配符与受限通配符	174
5.7.2	正则表达式的应用	128	7.7.5	子类泛型	175
5.8	数组实用类	131	7.7.6	泛型数组	177
5.8.1	复制数组	131	7.8	Java 增强容器特性	178
5.8.2	数组排序	132	7.9	综合实例	180
5.8.3	数组元素的查找	133	第 8 章	异常处理	183
第 6 章	Java 语言新特性	135	8.1	异常概述	183
6.1	枚举	135	8.2	异常处理机制	184
6.1.1	定义枚举类型	135	8.2.1	异常的捕获与处理	184
6.1.2	enum 构造方法	136	8.2.2	声明抛出异常子句	186
6.1.3	使用 EnumMap	137			

8.2.3 抛出异常	186	10.7 综合实例	233
8.3 自定义异常类	187	第 11 章 AWT 图形用户界面编程	238
第 9 章 Java 输入/输出系统	190	11.1 AWT 容器	238
9.1 字节流	190	11.1.1 Window 和 Frame	239
9.1.1 字节数组输入流	191	11.1.2 Panel	239
9.1.2 字节数组输出流	192	11.2 布局管理器	241
9.1.3 文件输入流	192	11.3 事件处理机制	243
9.1.4 文件输出流	193	11.3.1 AWT 事件与监听器	243
9.1.5 管道流	194	11.3.2 窗口事件	244
9.2 过滤流	195	11.3.3 鼠标事件	246
9.2.1 缓冲流类	196	11.3.4 键盘事件	247
9.2.2 数据流类	197	11.3.5 内部类实现监听器接口	249
9.2.3 PrintStream 类	199	11.3.6 类自身实现监听器接口	250
9.3 字符流	200	11.3.7 外部类实现监听器接口	251
9.3.1 转换流类	201	11.3.8 采用事件适配器	253
9.3.2 FileReader 和 FileWriter 类	202	11.4 在 AWT 中绘图	255
9.3.3 BufferedReader 和 BufferedWriter 类	203	11.4.1 Graphics 类	255
9.3.4 PrintWriter 类	204	11.4.2 AWT 绘图实现原理	257
9.4 标准 I/O	205	11.5 综合实例	259
9.4.1 重新包装标准 I/O	206	第 12 章 Swing 组件及应用	264
9.4.2 标准 I/O 流重定向	206	12.1 窗口 (JFrame)	264
9.5 随机访问文件类	207	12.2 Swing 基本组件	265
9.6 对象序列化	210	12.2.1 按钮 (JButton)	265
9.7 File 类	211	12.2.2 文本框 (JTextField)	267
9.8 综合实例	213	12.2.3 复选框 (JCheckBox) 和 单选按钮 (JRadioButton)	269
第 10 章 多线程	215	12.2.4 列表框 (JList)	272
10.1 线程的创建与启动	215	12.3 Swing 菜单程序设计	274
10.1.1 继承 java.lang.Thread 类	215	12.3.1 菜单条 (JMenuBar)	274
10.1.2 实现 Runnable 接口	217	12.3.2 弹出式菜单 (JPopupMenu)	277
10.2 线程的状态转换	218	12.3.3 页签面板 (JTabbedPane)	279
10.3 线程调度	219	12.3.4 工具栏 (JToolBar)	281
10.3.1 调整线程优先级	219	12.4 Swing 数据管理组件	283
10.3.2 线程让步	220	12.4.1 表格 (JTable)	284
10.3.3 合并线程	221	12.4.2 树状视图 (JTree)	286
10.4 后台线程	222	12.5 Swing 标准对话框	287
10.5 线程互斥	223	12.5.1 对话框 (JDialog)	287
10.5.1 临界区	223	12.5.2 消息框 (JOptionPane)	289
10.5.2 对象锁机制	225	12.5.3 文件对话框 (JFileChooser)	291
10.5.3 显式 Lock 对象	228		
10.6 线程同步	230		

12.6	综合实例	293	14.4.1	URL 类	327
第 13 章	Java 基础开发综合实习	297	14.4.2	URLConnection 类	329
13.1	设计思路	297	14.5	综合实例	331
13.2	汉诺塔上盘子模拟	297	第 15 章	Java 数据库编程	336
13.3	对象定位及盘子的存放	298	15.1	SQL 语言简介	336
13.4	创建汉诺塔及实现手工搬运盘子	299	15.2	JDBC 原理	337
13.5	自动搬运盘子	305	15.3	访问数据库	339
第 14 章	Java 网络编程	312	15.3.1	加载并注册数据库驱动	339
14.1	网络程序设计基础	312	15.3.2	建立到数据库的连接	340
14.1.1	TCP 与 UDP	312	15.3.3	执行数据库操作	341
14.1.2	端口和套接字	312	15.4	JDBC 操作 MySQL	342
14.2	TCP 网络编程	313	15.4.1	获取和加载驱动	342
14.2.1	TCP 通信模型	313	15.4.2	创建数据库和表	345
14.2.2	Socket 编程相关类	313	15.4.3	操作数据库	347
14.2.3	TCP 通信程序	316	15.4.4	事务处理	349
14.3	UDP 网络编程	319	15.4.5	存储过程	351
14.3.1	UDP 相关类	319	15.5	JDBC 操作其他数据库	353
14.3.2	UDP 通信程序	321	15.5.1	JDBC 操作 SQL Server	354
14.3.3	组播	324	15.5.2	JDBC 操作 Oracle	355
14.4	URL 网络编程	327	15.6	综合实例	356

第 2 部分 实 验

实验 1	Java 语言及编程环境	362	实验 8	Java 异常处理	377
实验 2	Java 语法基础	363	实验 9	Java 输入/输出系统	378
实验 3	Java 类与对象	365	实验 10	多线程	379
实验 4	Java 面向对象编程	368	实验 11	AWT 图形用户界面编程	380
实验 5	Java 常用类	371	实验 12	Swing 组件及应用	383
实验 6	Java 语言新特性	373	实验 13	Java 网络编程	386
实验 7	容器和泛型	374	实验 14	Java 数据库编程	392

第 3 部分 习 题 集

第 1 章	Java 语言及编程环境	399	第 8 章	异常处理	404
第 2 章	Java 语法基础	399	第 9 章	Java 输入/输出系统	404
第 3 章	Java 类与对象	402	第 10 章	多线程	404
第 4 章	Java 面向对象编程	402	第 11 章	AWT 图形用户界面编程	405
第 5 章	Java 常用类	403	第 12 章	Swing 组件及应用	405
第 6 章	Java 语言新特性	403	第 14 章	Java 网络编程	405
第 7 章	容器和泛型	403	第 15 章	Java 数据库编程	406

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，在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 Internet（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 诞生，于次年 1 月发布了第一个开发包 JDK 1.0，JDK 1.0 至 JDK 1.1（1997.2.18）奠定了 Java 在计算机语言中的地位。自 JDK 1.2（1998.12.8）起，Java 的应用平台逐步分化，演进成以下三个版本。

- **Java SE**——Java Standard Edition（标准版），主要用于普通 PC、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。在 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 公司将其更名为 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，甲骨文公司充分利用自身的技术和资源优势不遗余力地发展 Java，定期升级 Java SE 的版本，使 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中最流行的语言和平台之一，目前的最新版本是 Java SE 12.0。

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 和环境的限制，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 具有如下特点。

(1) 简单性。Java 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有几十 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。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，这就有效地防止了黑客使用“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软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 (以下简称 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。整个运行过程分为代码的装入、校验和执行这三步：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 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。自 Java 9 起，改用 G1 作为默认垃圾收集器 (取代原来吞吐优先的 CMS 垃圾收集器)，它引入分区 (Region) 机制，回收效率更高；Java 11 又开发了全新的垃圾收集器 ZGC，能够在数 TB 的堆上实现非常低的暂停时间……随着 Java 版本的升级，垃圾收集技术也越来越先进。

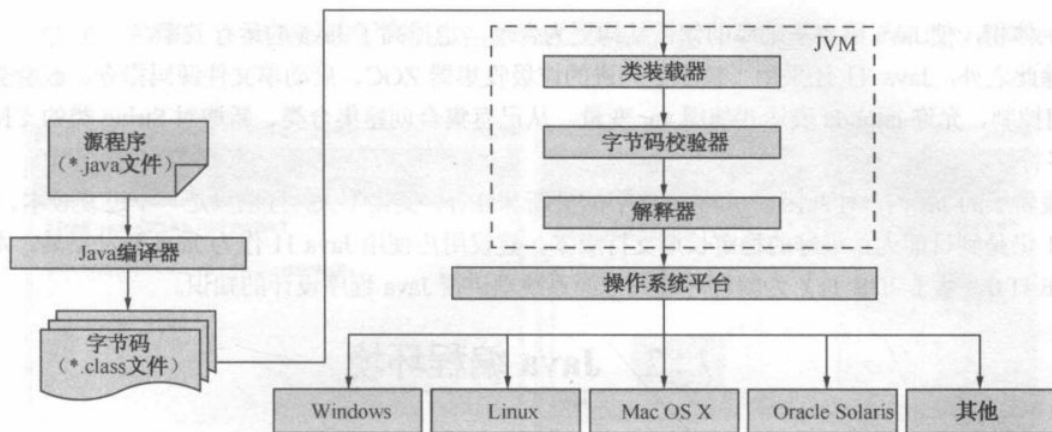


图 1.1 Java 程序执行机制

1.1.3 从 Java 5 到 Java 11

自 Java SE 5.0 起，Java 历经多代发展，在这期间有几次十分重大的升级。

1. Java 8

Java 8 是第一个重要的升级版，引入了如下新特性。

- **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。
- **新的流式 API**。流式 API（Application Programming Interface，应用编程接口）支持对数据执行管道操作，并针对 lambda 表达式做了优化。
- **简化的接口实现**。在 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，则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。
- **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。

2. Java 9

从这一版开始，甲骨文公司提高了 Java 升级的频率，每 6 个月固定发布一次版本升级，以期继续引领 Java 时代潮流。Java 9 的新特性如下。

- **引入模块系统 JPMS**。模块是包的容器，仅供依赖它的应用使用，使用模块机制让开发者构建和维护大型的库和应用程序变得更容易，也增强了系统的兼容性。
- **封装支持 HTTP 协议的工具有**。HTTP 客户端支持 HTTP2，同时改进 API，支持异步模式。
- **集合工厂**。优化了集合框架，提供类的工厂化构建方法，进一步增强了 Java 语言面向对象性能和编程灵活性。

3. Java 11

Java 10 是过渡版本，变化不大。Java 11 则是继 Java 8 之后官方公开发布的第一个长期支持（LTS）版本。此版本将之前 Java 9 和 Java 10 引入的新特性全部加以正式固化，同时又做了很多下述重要革新。

- **正式引入 HTTP 协议支持模块**。在 Java 9、Java 10 中被标记 incubator 的模块 jdk.incubator.httpclient，在 Java 11 中被标记为正式，改为 java.net.http 模块。
- **精简了 JDK 的结构和规模**。此次升级将 Java EE、CORBA、Pack200 工具及 Java FX 等非 Java SE 核心功能从 JDK 分离出去，以另外独立的库和包的形式提供，这么做极大地缩小了 JDK 的

体积,使 Java 语言基础库的体系结构更为合理,也提高了 Java 的运行效率。

除此之外,Java 11 还采用了技术更先进的垃圾收集器 ZGC、启动单文件源码指令、嵌套类的安全访问控制、允许 lambda 表达式使用 var 变量、从已有集合创建集合类、新增对 String 类的支持方法等诸多特性。

最新版的 Java 12 对 Java 11 做了一些小的更新和修补,变化不大,且也只是一个过渡版本。因此,Java 11 仍是到目前为止最好的稳定长期支持版本,建议用户使用 Java 11 作为 Java 开发工具。本书以 Java SE 11.0 (基于 JDK 11) 为编程环境,全面系统地讲授 Java 程序设计的知识。

1.2 Java 编程环境

1.2.1 JDK 11

要想编译和运行 Java 程序,离不开 Java 的编译和运行环境。甲骨文公司为 Java 提供一套原生的开发环境,通常称其为 JDK (Java SE Development Kits)。本书使用的 Java 11 基于 JDK 11。

1. 下载 JDK 11

可到甲骨文公司的官网下载 JDK 11: <https://www.oracle.com/technetwork/java/javase/downloads/index.html>, 单击“DOWNLOAD”按钮(如图 1.2 所示),在出现的下载页上先选中“Accept License Agreement”选项,然后单击“jdk-11.0.4_windows-x64_bin.exe”链接,下载对应 64 位 Windows 计算机的 JDK。

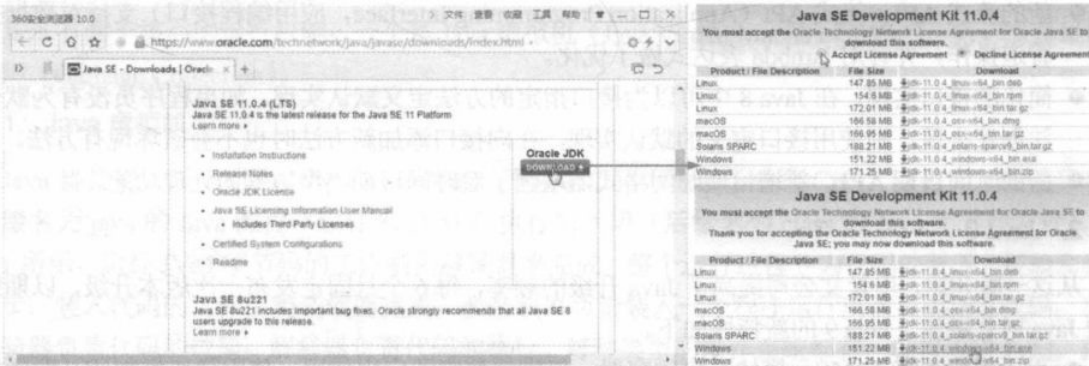


图 1.2 选择下载 JDK

2. 安装 JDK 11

下载完成后,得到可执行文件 jdk-11.0.4_windows-x64_bin.exe, 双击启动安装向导,按照提示完成安装,这里 JDK 的安装路径取默认的“C:\Program Files\Java\jdk-11.0.4”,如图 1.3 所示。

因为从 Java 11 开始,不再单独发布 JRE,所以安装后在 C:\Program Files\Java 下只生成一个名为 jdk-11.0.4 的文件夹目录。

3. 配置环境变量

JDK 安装完成后,还需要配置环境变量才可使用,下面介绍具体步骤。

1) 打开“环境变量”对话框

右击桌面上的“计算机”图标,选择“属性”命令,在弹出的“控制面板主页”中单击“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。

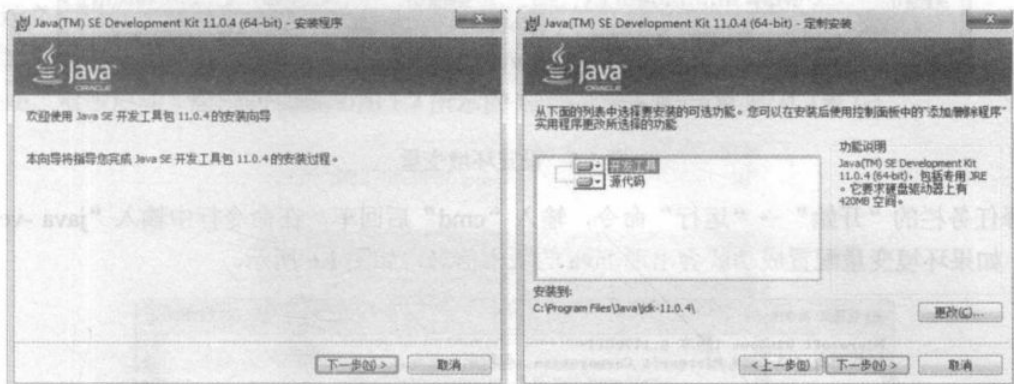


图 1.3 安装 JDK

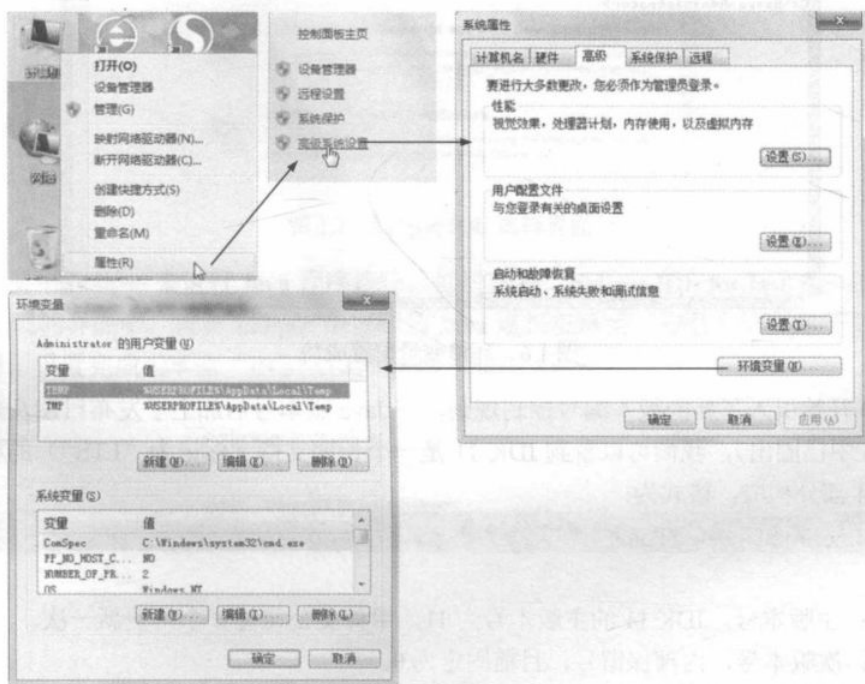


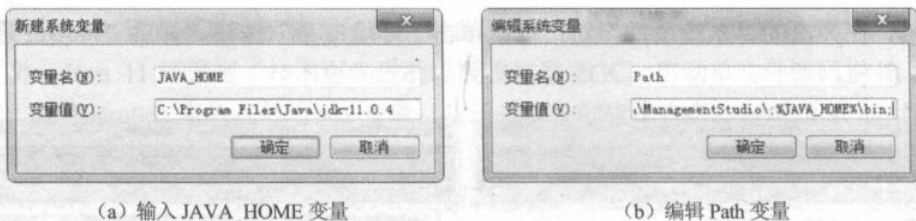
图 1.4 打开“环境变量”对话框

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”栏中输入“JAVA_HOME”，在“变量值”栏中输入 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”栏的字符串中加入路径“%JAVA_HOME%\bin;”，如图 1.5 (b) 所示，单击“确定”按钮。



(a) 输入 JAVA_HOME 变量

(b) 编辑 Path 变量

图 1.5 设置环境变量

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

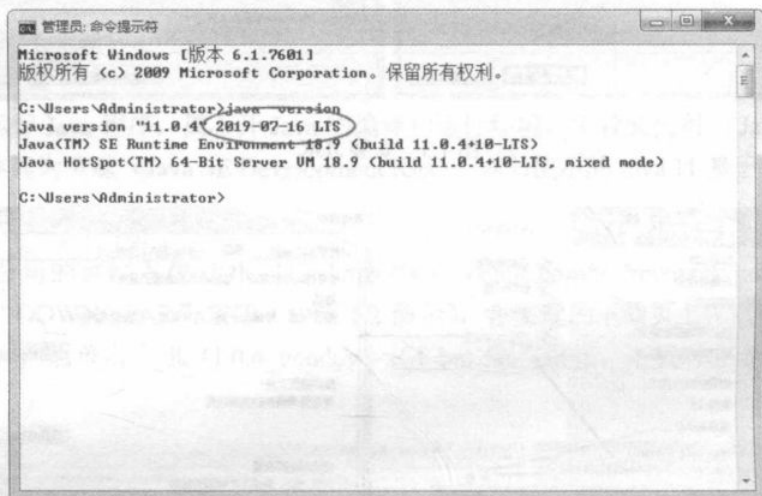


图 1.6 环境变量配置成功

从 Java 10 开始引入了新的版本编号控制规则，在 Java 版本号后加上了发布日期及是否为长期支持版（在图 1.6 中已圈出），我们可以看到 JDK 11 是一个长期支持（标记为“LTS”）的稳定版本。详细的版本号由 4 部分构成，格式为：

```
$Feature.$Interim.$Update.$Patch
```

其中，

- Feature: 主版本号，JDK 11 的主版本号为 11，甲骨文公司每 6 个月更新一次。
- Interim: 次版本号，内部保留号，目前固定为 0。
- Update: 更新补丁号，最初是 0，在主版本发布后的 1 个月增为 1，之后每 3 个月增加 1。例如，JDK 11.0.4 是在 2019 年 7 月 16 日发布的，可推出其上一版 11.0.3 是在 4 月发布的，其下一版 11.0.5 在 2019 年 10 月推出，依此类推。
- Patch: 紧急安全修复补丁编号，因 JDK 11.0.4 是稳定版，故不包含这一编号字段。

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，通常不这样做，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 的开发环境。

1. 安装 Eclipse 2019

最新版本的 Eclipse 是在 2019 年 6 月发布的 Eclipse 2019-06，其官方只提供安装器的下载，实际安装时还必须确保计算机处于联网状态以实时下载所需的文件。

(1) 先从 Eclipse 官网下载安装器：<https://www.eclipse.org/downloads/>，得到启动文件 eclipse-inst-win64.exe，双击启动安装器，出现如图 1.7 所示的 Eclipse IDE 选择界面。



图 1.7 Eclipse IDE 选择界面

因为我们要用 Eclipse 来进行 Java 程序设计，这里选择“Eclipse IDE for Java Developers”条目。

(2) 在出现的界面中，设置 Eclipse 所使用的 Java 虚拟机环境、安装目录及是否创建桌面和菜单快捷启动项等，设置完成后单击“INSTALL”按钮，如图 1.8 所示。



图 1.8 设置 Eclipse 安装选项

(3) 弹出的对话框让用户确认是否接受许可协议条款，单击“Accept Now”按钮表示接受，可开启安装进程，如图 1.9 所示，界面上出现绿色进度条。