



沈志宏 王松涛 周园春 编著 黄平山 审校

Java 6

从入门到精通



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

TP312/2861

2008

Java 6 从入门到精通

沈志宏 王松涛 周园春 编著

黄平山 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统而全面地介绍了 Java 语言的编程技术, 首先介绍 Java 编程的基础语法和基本操作知识, 包括 Java 基础知识、面向对象的编程思想、Java 语法基础、类与继承、Java 高级类特性、异常处理机制、常用 Java 类库、图形用户界面的设计与实现、GUI 事件处理、线程以及 Java 输入输出操作。然后从实用的角度出发, 列举并详细介绍了关于 Java 编程技术的高级话题, 包括如何在 Eclipse 中进行 Java 开发, 如何编写 Java 程序实现网络通信, 如何从 XML 文件中读取应用的配置信息, 如何进行关系型数据库的读写操作等。

本书具有由浅入深、通俗易懂和注重实践等特点, 适用于没有或者缺乏 Java 程序设计经验的初学者, 同时也适用于已掌握 Java 基础编程技术、需要提高 Java 实用技术的程序员, 也可以用做各类相关培训班的教材。

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 6 从入门到精通 / 沈志宏等编著. —北京: 电子工业出版社, 2008.5

ISBN 978-7-121-06217-9

I. J… II. 沈… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 036117 号

责任编辑: 王军花 易 昆

印 刷: 北京天竺颖华印刷厂

装 订: 三河市金马印装有限公司

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

北京市海淀区翠微东里甲 2 号 邮编: 100036

开 本: 787×1092 1/16 印张: 43.5 字数: 1100 千字

印 次: 2008 年 5 月第 1 次印刷

定 价: 76.00 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 zltz@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

Internet 技术是 20 世纪末人类科技史上最重要的成果之一，在人类生活中起着越来越重要的作用。Java 作为针对 Internet 应用开发的最流行的软件开发语言，得到了世界上绝大多数软件公司的全力支持，成为当今计算机技术的一个重要应用领域。

目前，Java 凭借其“Write Once, Run Anywhere”开放的、与平台无关的优势，迅速占领了包括网络通信、分布式存储计算、电子商务、嵌入式开发等应用领域。所以 IT 市场对 Java 程序员的需求也与日俱增，Java 程序员除了具有基本的 Java 编程能力之外，还必须具有更多的实际开发能力，以便更好更快地投入到应用系统的开发中去。针对广大 Java 学习者的实际需要，在通过对目前相关图书进行调研的基础上，我们特别策划编写了这本图书。

为了照顾到不同水平层次的读者，根据读者在各阶段对 Java 的学习需求和特点，本书从基础的 Java 编程语法开始，结合网络、数据库、XML 等技术要点，对 Java 的编程技术和应用进行了全面和系统的阐述。为了突出实用性，我们将在本书的后面列举和介绍相关的综合运用实例，并列举几种常用的流行开发技术，包括 JavaMail、Logging、Cache、ORM 等概念及开源软件的基本使用。

归结起来，本书具有如下特色：

知识完整而系统，适用面广。本书不仅有适合于初学者的基础语法介绍，同时还安排了包括网络、数据库、XML 等在内的高级 Java 编程技术，能够满足具有一定语法基础却急需提高、准备应用开发的 Java 程序员的需求。

注重编程实践。本书在前 12 章中每一章都配有代码实例，此外，在后面的章节中安排了“Java 编程综合实例”，以数据库连接池 DBCP 的实现作为实例，启发读者如何综合运用编程知识，完成 Java 高级应用的开发。

倡导正确的编程思想。本书并非按部就班地完成知识传授，而是在介绍知识的同时倡导正确的学习思想和方法，倡导 OOP 思想、流行的设计模式、不断地重构理念、开源精神，读者在阅读本书的同时，会学习到这些新的理念和方法。

本书内容丰富，按知识点的难易共分为 18 章。

本书引用了大量实例程序代码，许多理论知识都是以实例的演示说明和归纳总结来进行讲解的，从程序运行结果中得出抽象的理论，使读者易于理解与掌握。书中的程序实例许多都选自作者多年的 Java 程序设计和教学实践。本书在选材方面力图适当和实用，做到深入浅出，重点突出，因此本书具有很强的实用性和可读性。

本书不仅适合于没有或者缺乏 Java 程序设计经验的初学者，同时也适合于已掌握 Java 基础编程技术、需要提高 Java 实用技术的程序员，还可作为相关培训班的培训教材。

本书由平山工作室策划并组织编写，其中前 12 章由王松涛和周园春负责编写，后面部分由沈志宏负责编写，黄平山负责前期策划和全书的统稿工作，此外，胡鹏先生也参与了部分编写工作，并提出了很好的意见和建议，在此一并表示感谢。通过自己的努力工作，让读者在阅

读本书之后有所收获，有所启发，并能在 Java 编程开发的路上走得更远，这是我们最大的期望。

由于时间仓促，个别章节由于条件和水平所限难以深入介绍，加上 Java 编程技术日新月异的发展，因此书中难免有纰漏和欠缺之处，敬请广大读者批评指正。

目 录

第 1 章 初识 Java 技术	1	2.3.2 类定义中的关键字	26
1.1 Java 的起源、背景及前景	1	2.3.3 类主体	27
1.2 什么是 Java 技术	3	2.4 属性和方法	27
1.2.1 什么是 Java 技术	3	2.4.1 类属性	28
1.2.2 Java 编程语言的主要设计目标	5	2.4.2 类方法	30
1.3 第一个 Java Application 程序	6	2.4.3 构造方法	31
1.3.1 安装 Java 编译器和工具	6	2.5 对象的访问	33
1.3.2 创建一个源代码文件	7	2.5.1 用 new 运算符进行对象的 创建	34
1.3.3 编译源文件	8	2.5.2 用“.”运算符进行对象的 使用	34
1.3.4 运行程序	9	2.6 Java 接口和包	35
1.3.5 相关问题说明	9	2.6.1 Java 接口	35
1.4 创建第一个小应用程序	10	2.6.2 包	37
1.4.1 创建源文件	10	第 3 章 Java 语法基础	42
1.4.2 编译源文件	11	3.1 标识符、关键字、数据类型	42
1.4.3 运行程序	11	3.1.1 标识符	42
1.5 Java 虚拟机	11	3.1.2 Java 关键字	43
1.6 垃圾收集	12	3.1.3 基本 Java 数据类型	43
1.7 代码安全	12	3.1.4 自动装箱和解箱	49
1.8 Java 运行环境	13	3.2 变量、声明和赋值	54
1.9 Java 目录环境和 Java API 文档	13	3.2.1 声明一个变量	54
第 2 章 面向对象的编程	16	3.2.2 变量的作用域	55
2.1 面向对象编程	16	3.3 构造和初始化对象	57
2.1.1 面向对象的基本概念	17	3.4 类型转换与强制类型转换	58
2.1.2 对象和类	17	3.4.1 Java 的自动转换	58
2.1.3 面向对象的特征	19	3.4.2 不兼容类型的强制转换	59
2.2 面向对象的分析和设计	21	3.4.3 表达式中类型的自动提升	60
2.2.1 面向对象的分析与面向 过程的分析	21	3.5 表达式和流程控制	61
2.2.2 面向对象的分析和设计的 精髓	22	3.5.1 运算符	62
2.3 描述一个 Java 类	25	3.5.2 运算符优先级	76
2.3.1 Java 类实例的声明	26	3.5.3 流程控制	77
		3.5.4 特殊循环控制	87

3.6	数组	90	5.2.1	final 成员变量	160
3.6.1	数组的声明	90	5.2.2	final 方法	161
3.6.2	数组的创建和引用	91	5.2.3	final 类	162
3.6.3	数组的初始化	93	5.3	抽象类	162
3.6.4	多维数组	94	5.4	接口	166
3.6.5	复制数组	97	5.4.1	接口的定义	166
3.6.6	数组排序	97	5.4.2	接口的实现	167
3.6.7	foreach 与数组	99	5.4.3	接口的继承	168
3.7	可变参数	100	5.4.4	接口的应用	169
3.8	枚举类型	102	5.5	深入理解抽象类与接口	172
3.8.1	为什么使用枚举类型	102	5.6	内部类	176
3.8.2	枚举类型定义	103	5.7	匿名类	180
3.8.3	使用枚举值	104	第 6 章	异常	183
3.8.4	为枚举添加属性和方法	105	6.1	异常	183
3.8.5	枚举和集合	107	6.2	实例讲解	184
3.8.6	小结	109	6.3	异常处理	185
3.9	注解	110	6.3.1	try 和 catch 语句	186
3.9.1	元数据编程	110	6.3.2	嵌套 try 语句	190
3.9.2	注解的基本概念	110	6.3.3	调用栈机制	192
3.9.3	基本内置注解	111	6.3.4	throw 语句	192
3.9.4	定义注解	114	6.3.5	重新抛出异常	193
3.9.5	元注解	117	6.3.6	throws	194
3.9.6	元注解实例	120	6.3.7	finally 语句	195
第 4 章	继承	123	6.4	异常类	198
4.1	子类	123	6.4.1	Java 内置异常	199
4.2	super 关键字	131	6.4.2	定义异常类	201
4.3	重载 (Overload)	136	6.5	上机指导	202
4.4	覆盖 (Override)	137	第 7 章	常用 Java 类库介绍	208
4.5	多态与动态绑定	139	7.1	控制台输入输出	208
4.6	instanceof 运算符	144	7.2	Properties 类	210
4.7	定义和引用包	145	7.3	File 类	213
第 5 章	Java 高级类特性	153	7.3.1	目录	215
5.1	static 关键字	153	7.3.2	使用 FilenameFilter 接口	217
5.1.1	类 (static) 变量	153	7.3.3	listFiles() 方法	218
5.1.2	类 (static) 方法	155	7.3.4	创建目录	218
5.1.3	静态初始化程序	156	7.3.5	RandomAccessFile 类	218
5.1.4	静态类	158	7.4	Math 类	219
5.1.5	静态导入	158	7.5	字符串类	223
5.2	final 关键字	160	7.5.1	认识字符串	223

7.5.2	String 类	223	9.1.2	事件监听器	279
7.5.3	不可变字符串	227	9.2	J2SE 事件模型	280
7.5.4	StringBuffer 和 StringBuilder 类	228	9.2.1	层次模型 (JDK 1.0)	280
7.5.5	关于串处理的效率	230	9.2.2	委托模型 (JDK 1.0)	281
7.5.6	字符串上的高级操作	232	9.3	GUI 行为	283
7.5.7	格式化输出	237	9.3.1	事件类别	283
7.5.8	扫描输入	241	9.3.2	事件监听器类型	287
第 8 章	图形用户界面的设计与实现	243	9.3.3	事件监听器类型	288
8.1	图形用户界面的概述	243	9.4	适配器	298
8.1.1	AWT 是什么	243	9.4.1	什么是适配器	298
8.1.2	Swing 是什么	244	9.4.2	使用适配器举例	298
8.1.3	图形界面元素	244	9.5	系统托盘	302
8.2	使用 java.awt 包	245	第 10 章	线程	308
8.2.1	AWT 包中的类和关系	246	10.1	什么是线程	308
8.2.2	窗口类的层次结构	247	10.1.1	线程概念	308
8.2.3	使用 Frame 类构建应用程序界面	248	10.1.2	理解线程如何工作	309
8.2.4	丰富应用程序界面	250	10.2	Java 编程中的线程	310
8.3	使用控件	260	10.2.1	主线程	311
8.3.1	控件概述	260	10.2.2	线程的 3 个部分	313
8.3.2	标签	260	10.2.3	创建一个线程	313
8.3.3	按钮	262	10.2.4	启动线程	318
8.3.4	复选框	263	10.2.5	线程调度	318
8.3.5	复选框组	264	10.3	线程的基本控制	327
8.3.6	选择框	265	10.3.1	终止一个线程	327
8.3.7	列表框	266	10.3.2	测试一个线程	330
8.3.8	滚动条	267	10.3.3	线程优先级	333
8.3.9	文本框	268	10.3.4	线程组	338
8.3.10	文本区	269	10.4	J2SE SDK 中的线程控制	341
8.4	使用控件布局管理	270	10.4.1	线程同步	341
8.4.1	边界布局管理器	270	10.4.2	线程间通信	344
8.4.2	流式布局	272	10.4.3	线程挂起、恢复和终止	348
8.4.3	网格布局管理器	273	10.4.4	线程死锁	354
8.4.4	卡片布局管理器	274	10.5	J2SE 并发控制包	358
8.4.5	控件布局多样化	276	10.5.1	并发集合 (Collection)	358
第 9 章	GUI 事件处理	278	10.5.2	Callable 接口	365
9.1	什么是事件	278	10.5.3	新任务执行架构	366
9.1.1	事件源	278	10.5.4	新线程同步机制	368
			10.5.5	线程池	370
			10.5.6	其他主要类	376

10.5.7	多线程实例	386	12.5	泛型与数据类型转换	432
第 11 章	Java 输入输出操作	395	12.5.1	消除类型转换	432
11.1	输入输出流原理	395	12.5.2	具有自动装箱和解箱的功能	433
11.2	字节流	396	12.5.3	受限泛型	433
11.2.1	InputStream 类	397	12.6	擦除	434
11.2.2	OutputStream 类	397	12.7	Java 中的泛型和 C++ 模板的比较	436
11.2.3	FileInputStream 类	398	12.8	泛型的一些局限性	436
11.2.4	FileOutputStream 类	400	第 13 章	使用 Eclipse 进行 Java 开发	438
11.2.5	ByteArrayInputStream 类	401	13.1	Eclipse 是什么	438
11.2.6	ByteArrayOutputStream 类	402	13.1.1	Eclipse 介绍	439
11.2.7	过滤字节流类	404	13.1.2	Eclipse 的安装	440
11.2.8	BufferedInputStream 类	404	13.1.3	了解 Eclipse 的工作台	441
11.2.9	BufferedOutputStream 类	406	13.2	在 Eclipse 里面编辑 HelloWorld	443
11.2.10	PushbackInputStream 类	406	13.2.1	创建一个项目	443
11.2.11	SequenceInputStream 类	408	13.2.2	创建一个类	444
11.2.12	PrintStream 类	409	13.2.3	完成 main() 方法	449
11.3	字符流	410	13.2.4	即时语法检查	449
11.3.1	Reader 类	410	13.3	在 Eclipse 里面运行 Hello World	451
11.3.2	Writer 类	410	13.3.1	运行 Hello World	451
11.3.3	FileReader 类	411	13.3.2	调试 Hello World	452
11.3.4	FileWriter 类	412	13.4	对 HelloWorld 进行单元测试: JUnit	453
11.3.5	CharArrayReader 类	413	13.4.1	验证这个 mcd 算法	454
11.3.6	CharArrayWriter 类	414	13.5	对 HelloWorld 进行版本管理: CVS	458
11.3.7	BufferedReader 类	415	13.5.1	版本与版本管理	458
11.3.8	BufferedWriter 类	416	13.5.2	对 HelloWorld 进行版本管理	459
11.3.9	PushbackReader 类	417	13.5.3	提交 (commit)	463
11.3.10	PrintWriter 类	418	13.5.4	更新 (update)	464
11.4	序列化	418	13.5.5	版本冲突的解决	466
11.4.1	序列化的方法	419	13.6	使用 Eclipse 插件	467
11.4.2	序列化中的注意事项	420	13.6.1	Eclipse 插件介绍	467
第 12 章	泛型	422	13.6.2	Eclipse 插件安装	468
12.1	为什么使用泛型	422	第 14 章	Java 网络编程	472
12.2	简单泛型	424			
12.2.1	定义简单泛型类	424			
12.2.2	泛型和子类继承	426			
12.3	泛型方法	427			
12.4	泛型通配符	428			
12.4.1	有限制的通配符	430			
12.4.2	无限制通配符	431			

14.1 网络与 Internet	472	15.3.3 XML 数据的检索	541
14.1.1 WWW 服务	472	15.3.4 XML 数据的修改	544
14.1.2 网络编程基础	473	第 16 章 Java 数据库开发	547
14.2 基于 URL 的高层次 Java 网络编程	475	16.1 数据库概述	547
14.2.1 统一资源定位器 URL	475	16.1.1 关系型数据库	547
14.2.2 URL 类	476	16.1.2 SQL 语句	548
14.2.3 从 URL 读取网络资源	479	16.1.3 SQL Server 2005 的安装 和配置	548
14.2.4 通过 URLConnetction 连接 万维网	481	16.2 JDBC 编程	552
14.3 创建 TCP Socket 应用程序	484	16.2.1 数据库编程接口	552
14.3.1 TCP/UDP 协议	484	16.2.2 JDBC 概述	554
14.3.2 Client/Server 模式	484	16.2.3 JDBC 操作流程	555
14.3.3 Socket 类	485	16.2.4 使用 Statement 完成查询	565
14.3.4 简单 Client/Server 程序 实例	488	16.2.5 ResultSet 的概念与属性	571
14.3.5 交互 Client/Server 程序 实例	493	16.2.6 使用 ResultSet 操纵数据	573
14.3.6 多客户 Client/Server 程序 实例	497	16.3 JDBC 数据类型	578
14.4 使用 UDP 数据报和套接字	500	16.3.1 SQL 类型	578
14.4.1 数据报 DatagramPacket	500	16.3.2 数据类型映射	579
14.4.2 数据报套接字 Datagram Socket	502	16.3.3 类型映射的场合	581
14.4.3 UDP 聊天程序实例	502	16.3.4 LOB (大对象)	584
14.4.4 组播套接字 Multicast Socket	509	16.4 获取元数据	587
第 15 章 Java XML 开发	514	16.4.1 结果集元数据	587
15.1 XML 基础	514	16.4.2 数据库元数据	588
15.1.1 XML 概述	514	第 17 章 Java 编程综合实例	592
15.1.2 XML 语法	515	17.1 使用 JDBC 实现对象的 持久化	592
15.1.3 XML 文档的描述	518	17.2 重构 (refactor) UserDao	601
15.2 Java 与 XML 解析	520	17.3 使用连接池 DBCP 1.0	611
15.2.1 DOM	520	17.4 使用容器对象实现 DBCP 1.1	617
15.2.2 SAX	521	17.5 DBCP 1.2: 加上检测线程 PoolMonitor	627
15.2.3 JAXP	521	17.6 DBCP 1.3: 从 XML 中加载 连接池的配置	632
15.3 使用 JDOM 操作 XML	531	17.7 DBCP 1.4: 实现 JDBC 3.0 接口	639
15.3.1 JDOM 初体验	532	17.8 TODO: DBCP 2.0	641
15.3.2 XML 文件的创建	537	第 18 章 实用 Java 技术指南	645
		18.1 邮件 JavaMail	645
		18.1.1 JavaMail 概述	645

18.1.2	JavaMail 初体验	646	18.3.1	cache 概述	667
18.1.3	SMTP 身份验证	650	18.3.2	EHCache 初体验	669
18.1.4	发送 HTML 邮件	653	18.3.3	使用外部存储	672
18.2	日志 Logging	656	18.3.4	使用配置文件	674
18.2.1	日志概述	657	18.4	OR 映射 ORM	675
18.2.2	Log4j 初体验	659	18.4.1	ORM 概述	675
18.2.3	定义日志级别	664	18.4.2	Hibernate 初体验	677
18.2.4	使用配置文件	665	18.4.3	使用查询	683
18.3	对象池 cache	667			

第1章 初识 Java 技术

本章目标

Internet 技术是上世纪末人类科技史上最重要的成果之一，在人类生活中正起着越来越重要的作用。Java 作为针对 Internet 应用开发的最流行的软件开发语言，得到了世界上绝大多数软件公司的全力支持，是当今计算机技术发展的重要应用领域。

本章将对 Java 技术的历史、背景及其技术概要做一个简要介绍，使读者对 Java 语言有一个整体的了解。

重点与难点

- 初识 Java
- Java Application 的界面
- 获得帮助
- 新建和运行一个简单的 Java 应用程序

1.1 Java 的起源、背景及前景

Java 语言起源于 1991 年，那时 Sun 公司制定了一个名为 Green 的内部研究计划，目标是设计一种小型计算机语言，可用于消费类电器设备（如交互式电视、烤面包箱等）的软件开发。由于这些设备没有多少电力或内存，因此语言必须短小，并能产生非常严密的代码。此外该语言不能依赖于任何一种 CPU 体系结构。该项目的领导者 James Gosling 因看到窗口的橡树而得到灵感决定将这种程序语言命名为 Oak（橡树），这就是 Java 语言的前身。后来发现 Oak 已是 Sun 公司的另一个语言的注册商标，某日当 Green 中的成员喝咖啡时，因其咖啡产自爪哇（太平洋上一个盛产咖啡的岛屿），因此将该语言更名为 Java。

由于市场对智能型家电的需求并不如预期所料，Green 计划失败，也面临被取消的命运。与此同时，即 1993 年，Internet 上的 WWW 开始爆炸式的蓬勃发展。WWW 的关键是浏览器，需要一个不依赖于任何硬件和软件平台，并且可靠安全有交互功能的浏览器。在 Sun World 访谈中，Gosling 讲到，在 1994 年中 Java 开发人员意识到“我们可以构建一个非常精致的浏览器。在目前主流客户、服务器中几乎没有这样的产品。该浏览器具有它们已经实现的许多神奇功能：结构中性和实时、可靠、安全。这些问题在工作站领域中并不非常重要。所以我们构造了一个浏览器”。由此决定了用 Java 开发 WebRunner 项目。1995 年 WebRunner 推出，在业界引起了巨大的轰动，后来演变成 HotJava 浏览器。Java 的迅猛传播和发展也由此拉开了序幕。1996 年初 Java 1.0 正式发布。

如今 Java 技术已经进入了主流计算模式，并被广泛应用于科学研究、军事工业、信息技术、电子、航空航天、机械制造、科学测绘等各个科学领域，发挥着巨大的作用。

目前全球有 90% 的大型企业使用 Java 开发自己的信息系统；多家公司从 Sun 公司获得了 Java 技术许可证；Java 开发者阵营拥有几百万的会员。Java 从 1995 年的一个编程语言，发展到了今天可驾驭从嵌入式、工作站到大型企业级应用的 Java 平台，其发展速度和生命力是惊人的。自推出以来，Java 先后经历过多次的发展变化（如表 1-1 所示），始终随着计算机科学的进步而不断发展变化。

本书将以图文相辅的方式，用简洁严谨的描述为读者全面系统地介绍 Java 语言的基本特性，如表 1-1 所示。

表 1-1 Java 的发展历程

发布日期	版本信息
1995 年 05 月 23 日	Sun 发布了 Java 的 Alpha 1.0a2 版
1996 年 01 月 23 日	Java 第一个开发包 JDK 1.0.2 发布
1996 年 05 月 29 日	Servlet/JavaBean/Media
1996 年 10 月 29 日	Java Card API 发布
1997 年 02 月 18 日	JDK 1.1 发布
1997 年 02 月 28 日	Netscape 宣布支持 Java
1997 年 04 月 02 日	EJB 发布
1997 年 09 月 12 日	JDK 1.1.4 发布，代号为 Sparkler（宝石）
1997 年 12 月 03 日	JDK 1.1.5 发布，代号为 Pumpkin（南瓜）
1998 年 03 月	JFC/Swing 开始发售
1998 年 04 月 24 日	JDK 1.1.6 发布，代号为 Abigail（阿比盖尔）
1998 年 06 月 03 日	Visa 集团采用 Java Card 技术
1998 年 09 月 28 日	JDK 1.1.7 发布，代号为 Brutus（布鲁图）
1998 年 12 月 04 日	Sun 发布了 Java 2 平台，J2SE 1.2 发布，代号为 Playground（运动场）。并且开始分为 J2ME、J2SE、J2EE 三大块
1998 年 12 月 08 日	JCP 正式化
1999 年 03 月 04 日	XML 支持
1999 年 03 月 30 日	J2SE 1.2.1 发布，没有命名代号
1999 年 04 月 08 日	JDK 1.1.8 发布，代号为 Chelsea（切尔西）
1999 年 06 月	Sun 重新组织 Java 平台，并将 Java 企业级应用平台作为发展方向
1999 年 06 月 02 日	Java Server Page
1999 年 07 月 08 日	J2SE 1.2.2 发布，代号为 Cricket（蟋蟀）
1999 年 12 月 08 日	J2EE 开始发售
2000 年 05 月 08 日	J2SE 1.3.0 发布，代号为 Kestrel（美洲红笋）
2001 年 05 月 17 日	J2SE 1.3.1 发布，代号为 Ladybird（瓢虫）
2002 年 02 月 13 日	J2SE 1.4.0 发布，代号为 Merlin（灰背隼）
2002 年 09 月 16 日	J2SE 1.4.1 发布，代号为 Grasshopper（蚱蜢）
2003 年 06 月 26 日	J2SE 1.4.2 发布，代号为 Mantis（螳螂）
2004 年 09 月 29 日	J2SE 1.5.0 发布，并改名为 J2SE 5.0，代号为 Tiger（老虎）
2006 年 12 月 11 日	J2SE 6.0 发布，代号为 Mustang（野马）

要有效地完成书中练习，建议读者采用以下系统配置：

- Intel® Pentium® 或更高版本，800 MHz 或更快的处理器或兼容产品；
- Microsoft® Windows® XP（专业版、家庭版或 Tablet PC 版本）、Windows 2000 或 Windows NT® 4.0（SP6a 或更高版本）；
- 最低 128MB RAM（建议）；
- 最低需要安装 100 MB 可用磁盘空间；
- Microsoft® Internet Explorer 6.0；
- 鼠标、跟踪球或兼容的指示设备；
- CD-ROM 驱动器或者 DVD 驱动器。

1.2 什么是 Java 技术

Java 技术由语言、运行环境和工具集三部分组成，本节介绍 Java 技术的概念组成和 Java 编程语言的总体设计目标。

1.2.1 什么是 Java 技术

Java 是一个广泛使用的网络编程语言，它是一种新的计算概念。Java 技术可以从以下几个方面加以理解：

- 一种编程语言
- 一个开发平台
- 一个应用环境
- 一个部署环境

Java 语言具有如下特点：

- 简单性（Simple）

C/C++的强大功能是众所周知的，即使在众多程序语言的竞争下，C/C++仍旧在开发软件的程序语言占有相当的地位，然而学习或者使用 C/C++并不容易。很多时候开发人员并不需要使用 C/C++的一些功能，但为了使用 C/C++却需付出相当的学习成本、开发成本或者维护成本。

设计 Java 的成员们以自己常年的开发经验判断，在使用一些 C/C++的功能时，所得到的坏处可能多于好处（尤其是被一些没有经验的开发人员使用时），所以 Java 舍弃了 C/C++中一些较少使用、难以掌握或可能不安全的功能，像指针（Point）、运算符重载（Operator Overloading）和多重继承（Multiple Inheritance）等。

Java 除去一些 C/C++中复杂或不安全的功能，并在许多开发人员常使用的特性上加以简化而使之易于使用。例如字符串在 Java 中的处理就更为简单。Java 在设计时参考了许多 C/C++的语法与特性，使得学习过 C/C++的开发人员可以在短时间内了解如何使用 Java。

- 跨平台性（Cross-Platform）和解释性（Interpreted）

Java 语言的跨平台性是通过其语言的解释性来实现的。一个程序在一个平台上要能够运行，必须先解释为该平台所能理解的原始机器语言，但最大的问题在于每个平台所理解的机器语言各不相同。例如 Windows 操作系统理解的机器语言就与 Linux 不同，专为 Windows 操作系统所编写并编译好的程序，无法直接拿来在 Linux 操作系统上运行。

为了解决在不同平台间运行程序的问题，Java 程序在进行编译时，并不直接编译为与平台相对应的原始机器语言，而解释为与系统无关的“字节代码”。为了运行 Java 程序，运行的平台上必须安装有 Java 虚拟机（JVM，Java Virtual Machine），JVM 等于是为了 Java 程序所虚拟的环境。当运行 Java 程序时，Java 实时编译器（JIT，Just In Time Compiler）会将字节代码编译为目标平台所接受的原始机器语言，通过 JVM Java 程序可以在不同平台上运行。

- 适合分布式计算环境

Java 本身就是通过网络而产生的，它的许多功能与应用都与网络相关，从最初的 Applet、简化的 Socket、交互的 JSP/Servlet 网络程序到现在的 Web Service 等，都说明 Java 比较适合分布式环境。事实上，Java 应用最多的领域也正是分布式环境下的相关应用。

- 面向对象（Object-Oriented）

面向对象分析（Object-Oriented Analysis）是分析问题的一种方式，面向对象设计（Object-Oriented Design）是使用面向对象的思考方式来设计问题的解决方案。但是，要了解什么是面向对象，以及学会使用面向对象的方式进行设计与解决问题，并不是简单的事情，事实上让你了解面向对象也是这本书的目标之一。

具体面向对象的相关知识在后面的章节中会逐渐以实际的例子展开。Java 语言让你可以用面向对象的思考方式来设计并编写程序。面向对象的好处之一，就是可以让你设计出可重用的组件，或者是直接使用别人编写好的 Java 组件，并使开发出来的软件更具弹性且容易维护。

- 安全性（Security）

Java 加入了垃圾收集（Garbage Collection）机制，让开发人员无需担心对象资源的回收问题，异常处理架构也让开发人员可以掌握程序中各种突发的异常状况。另外，像 synchronized、final 等存取关键词的使用，目的都在于加强 Java 程序的安全性。

- 健壮性（Robust）

Java 将 C/C++ 中一些强大但不太容易掌握的功能去除了，比如指针（Point），去除这些功能的目的是让 Java 在使用时更加简单，编写出来的程序更为健壮。

- 高效、多线程、动态性和可移植性（Portable）

Java 至今主要发展成为 3 个领域的应用平台（如图 1-1 所示）：Java 2 Platform, Standard Edition (J2SE)、Java 2 Platform, Enterprise Edition (J2EE) 与 Java 2 Platform, Micro Edition (J2ME)。

- Java 2 Platform, Standard Edition (J2SE)

J2SE 是 Java 各应用平台的基础，也是本书主要的介绍对象。J2SE 主要用于工作站环境，其中包括编译程序和运行系统，利用它可以编写、部署和运行 Java 应用程序，是 PC 的 Java 标准平台。

J2SE 可以分为 4 个主要部分：JVM、JRE、JDK 与语言。

为了能运行所编写好的 Java 程序，平台上必须有 JVM。JVM 包括在 Java 运行环境（JRE，Java Runtime Environment）中，所以为了要运行 Java 程序，必须取得 JRE 并进行安装。如果要开发 Java 程序，则必须取得 JDK（J2SE Development Kits），JDK 包括了开发过程中所需要的一些工具程序，像 javac、java、appletviewer 等工具程序（关于 JDK 的安装与使用介绍，稍后说明）。

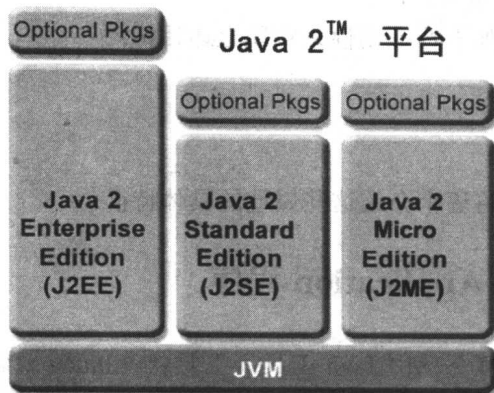


图 1-1 Java 2 平台

Java 语言只是 J2SE 的一部分，除了语言之外，Java 最重要的就是它提供了庞大而且功能强大的 API 类库，提供像字符串、数据输入/输出、网络组件等功能。开发者可以使用这些 API 作为基础来进行程序开发，而不需重复开发功能相同的组件。

● Java 2 Platform, Enterprise Edition (J2EE)

J2EE 以 J2SE 为基础，定义了一系列的服务、API、协议等，适用于开发分布式、多层次 (Multi-tiered)、以组件为基础、以 Web 为基础的应用程序，所以 J2EE 主要用于构建企业级应用，是分布式企业软件组件架构的规范，旨在简化和规范多层分布式企业应用系统的开发和部署。

整个 J2EE 的体系是庞大的，包括比较常用的技术像 JSP、Servlet、EJB (Enterprise JavaBeans)、RMI (Remote Method Invocation) 等。

● Java 2 Platform, Micro Edition (J2ME)

J2ME 是 Java 平台版本中最小的一个，目的是作为小型数字设备上开发及部署应用程序的平台，像用于嵌入式消费型产品等。最为人熟悉的设备如移动电话、PDA、汽车导航系统或其他无线设备等。

1.2.2 Java 编程语言的主要设计目标

Java 编程语言的主要设计目标是：

- 成为一种易于编程的语言，可以：
 - 消除其他语言在诸如指针运算和存储器管理方面影响代码健壮性的缺陷。
 - Java 语言采用了面向对象技术。
 - 能够使代码尽可能简洁、合理。
- 提供了一个解释环境，好处如下：
 - 在开发速度上，减少程序编译—链接—装载—测试周期的时间。
 - 在代码的可移植性方面，可以做到一次编写，到处运行。
- 提供多线程的环境。
- 提供在程序运行期内通过允许下载代码模块，动态改变程序的方法。

- 确保代码模块被安全载入。

Java 技术体系结构通过如下特征属性来实现上述目标：

- Java 虚拟机
- 垃圾收集
- 代码安全

在随后的章节中，我们将逐个介绍这些属性的具体特征。

1.3 第一个 Java Application 程序

本章介绍如何在各种环境下使用 Java 工具，主要以 Windows 操作系统为主。需要从 Sun 网站下载 Windows 平台的正式 JDK 版本包，并安装。

提示

对于各个平台上的 Java 版本，最好定期地访问 Java 主页，查看是否又发行了关于本平台的新版本。Java 主页是 <http://java.sun.com>。

注意

下载的 JDK 文件均是免费的，Java 语言规范是公开的，从 Sun 公司的网站上可以找到。

1.3.1 安装 Java 编译器和工具

本书的所有代码基于（J2SE 6.0）开发，没有该环境的读者可以从 Java 主页 <http://java.sun.com> 下载适合 Windows 平台的 J2SE 6.0 安装文件，如：`jdk-6-windows-i586.exe`，双击开始安装，出现安装进度界面，如图 1-2 所示。

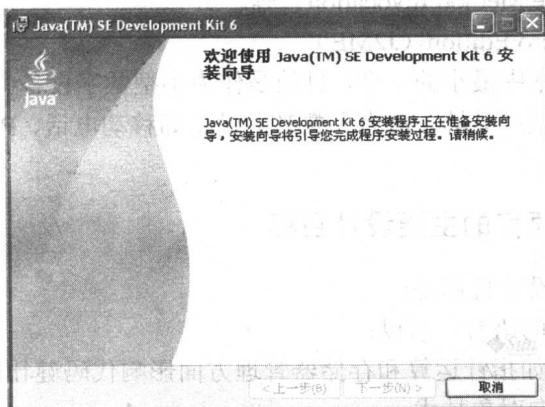


图 1-2 JDK 安装程序界面

随后会提示读者选择安装类型，如图 1-3 所示，单击【下一步】按钮，直至“安装完成”对话框出现。程序会默认安装到“Windows 安装盘符:\Program Files\Java\jdk1.6.0”路径下（如果该盘没有足够的空间，而且有其他磁盘分区，可改为其他驱动器，如 D:或 E:。）

下一节将创建一个简单的 Java Application 程序实例，通过图文并茂的方式向读者展现 Java 应用程序的创建过程。