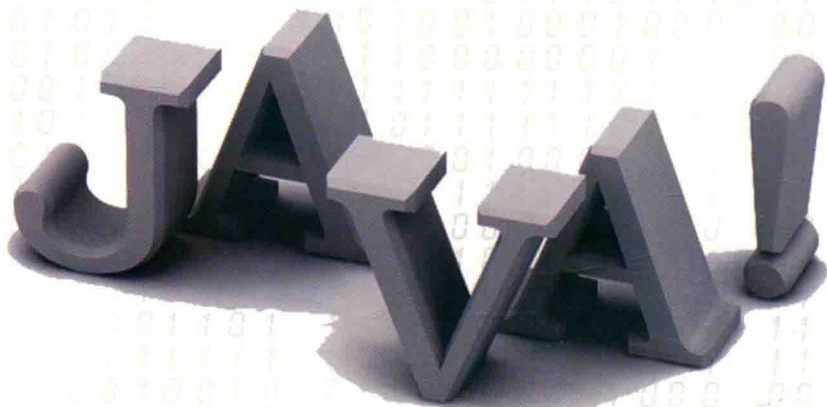


Java

语言程序设计与应用

徐俊武 编著

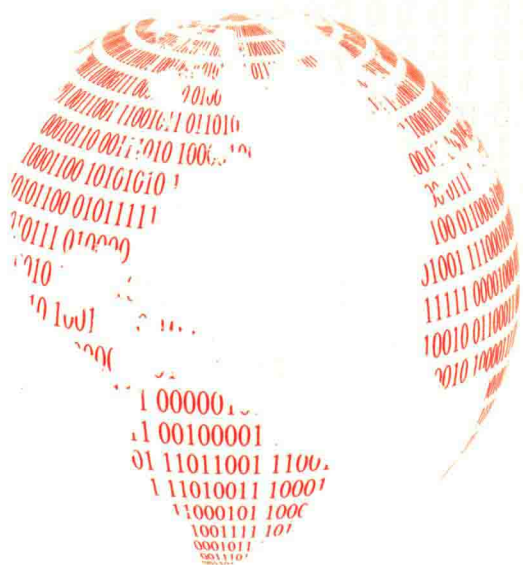


武汉理工大学出版社
Wuhan University of Technology Press



责任编辑：雷红娟

封面设计：博壹臻远



武汉理工大学出版社

网 址：<http://www.wutp.com.cn>

地 址：武汉市洪山区珞狮路122号

邮 编：430070

电 话：(027) 87532928 87391631

87785758

传 真：(027) 87165708



ISBN 978-7-5629-5869-7



9 787562 958697 >

定价：89.00元

Java 语言程序设计与应用

徐俊武 编著

武汉理工大学出版社

· 武 汉 ·

内 容 提 要

本书从初学者的角度出发,从 Java 基础知识开始,到面向对象程序设计,最终完成对应用程序的开发。全书共分为 8 章,分别介绍了 Java 语言基础、数组、Java 对象和类、Java 继承和接口、Java 用户界面控件、流与文件操作、Java 异常和调试等内容。本书注重面向对象的编程实践和问题分析能力的训练,使用大量例题来帮助理解和应用知识。本书通俗易懂,便于自学,每一章都配有应用举例,以帮助读者理解该章节的主要内容。

本书可供高校 Java 开发相关课程使用,也可作为 Java 培训班的培训教材。

图书在版编目(CIP)数据

Java 语言程序设计与应用/徐俊武编著. —武汉:武汉理工大学出版社,2019.1
ISBN 978-7-5629-5869-7

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

中国版本图书馆 CIP 数据核字(2018)第 279901 号

项目负责人:王兆国

责任编辑:雷红娟

责任校对:夏冬琴

封面设计:博壹臻远

出版发行:武汉理工大学出版社

社 址:武汉市洪山区珞狮路 122 号

邮 编:430070

网 址:<http://www.wutp.com.cn>

经 销:各地新华书店

印 刷:武汉中远印务有限公司

开 本:787×1092 1/16

印 张:15.5

字 数:295 千字

版 次:2019 年 1 月第 1 版

印 次:2019 年 1 月第 1 次印刷

定 价:89.00 元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。

本社购书热线电话:027-87515778 87515848 87785758 87165708(传真)

· 版权所有 盗版必究 ·

前 言

计算机科学是当今世界上发展最快、应用最广泛的科学技术之一。由于 Java 在面向对象的程序设计及其在网络开发中的广泛应用,引起了广大学生、计算机应用开发者的学习兴趣。不少学校已把 Java 程序设计作为计算机专业和相关专业的必修课程,为此撰写了本书。

本书共分为 8 章。第 1 章介绍了 Java 程序设计的基本概念、发展历程。第 2 章介绍了关键字、数据类型、运算符、表达式,以及各种相关类型的常量、变量的说明,程序设计的控制机构。第 3 章介绍了数组的概念,一维数组的定义、初始化以及常见的应用,二维数组的定义与使用。第 4 章介绍了 Java 语言对面向对象的实现机制,也就是类和方法,还给出了正确运用 OOP 设计的建议。第 5 章介绍了继承和接口。继承可以利用已有的类并根据需要来修改它,通过接口可以充分了解 Java 完全面向对象的程序设计能力。第 6 章介绍了用户界面控件。通过 Swing 工具包能够建立跨平台的图形界面,学习建立按钮、滑块、列表框、菜单等控件的使用方法。第 7 章介绍了流与文件的基本概念、Java 标准的输入/输出流和对象、文件流 I/O 及常用函数。第 8 章介绍了异常和调试,包括异常的基础知识、异常的处理、用户自定义异常。

本书的内容源于作者多年的教学实践经验及学习心得,并吸收了国内外同行的研究成果。本书可作为工程技术人员及高校学生的参考书。

由于计算机科学技术发展迅速,Java 程序设计的内容、方法和手段日新月异,加之编者水平有限,书中不足之处在所难免,敬请读者批评指正,以便在今后进一步完善。

编著者

2018.4

目 录

1 Java 介绍	1
1.1 了解 Java	1
1.1.1 Java 的诞生和发展	1
1.1.2 Java 三大平台	8
1.1.3 Java 语言的特点	8
1.1.4 Java 虚拟机	11
1.2 JDK	12
1.2.1 Windows 系统环境的 JDK 的安装与设置	12
1.2.2 编译和运行 Java 程序	17
2 Java 语言基础	19
2.1 语言成分	19
2.1.1 标识符和关键字	19
2.1.2 基本数据类型	21
2.1.3 变量与常量	24
2.1.4 运算符与表达式	26
2.2 字符串	35
2.2.1 字符串的基本数据类型特性	35
2.2.2 字符串的类特性	37
2.3 控制语句	39
2.3.1 流程控制结构	39
2.3.2 顺序结构语句	41
2.3.3 选择语句	42
2.3.4 循环语句	52
2.3.5 转移语句	63
3 数组	70
3.1 数组的概念	70
3.1.1 数组说明	70

3.1.2	创建数组	71
3.2	数组边界	72
3.3	数组元素引用	73
3.4	数组的传递	74
3.5	数组查找	76
3.6	数组排序	78
3.6.1	选择排序	78
3.6.2	冒泡排序	81
3.6.3	插入排序	83
3.7	多维数组	84
3.7.1	多维数组的定义	84
3.7.2	多维数组的初始化	85
3.7.3	多维数组的引用	87
3.7.4	数组复制	89
4	Java 对象和类	91
4.1	面向对象程序设计简介	91
4.1.1	面向过程的程序设计	91
4.1.2	面向对象的程序设计	91
4.2	创建自己的类	92
4.2.1	类的定义	92
4.2.2	成员变量的声明和初始化	93
4.2.3	成员方法的声明	94
4.2.4	成员方法的重载	95
4.2.5	构造方法	96
4.3	静态成员	98
4.3.1	类变量的声明和初始化	98
4.3.2	类方法	100
4.4	几个 Java API 中的标准类	101
4.4.1	Math 类	101
4.4.2	字符串类 StringBuffer	102
4.4.3	Date 类	104
4.5	对象构造	106
4.5.1	对象的创建	106
4.5.2	对象成员的使用	107

4.5.3	对象的清除	109
4.6	包	110
4.6.1	包的创建	110
4.6.2	Import 关键字	112
4.6.3	package 的目录结构	112
5	Java 继承和接口	114
5.1	扩展类	114
5.1.1	类继承	114
5.1.2	@Override 标记	116
5.1.3	选择基类访问属性	116
5.1.4	多态	117
5.1.5	多级继承	118
5.1.6	抽象类	118
5.1.7	通用超类	119
5.1.8	接收数目可变的参数的方法	121
5.1.9	转换对象	122
5.2	Class 类	123
5.2.1	Java 中的 Class 类	123
5.2.2	如何得到 Class 的对象	123
5.2.3	Class 类的常用方法	124
5.2.4	Class 的一些使用技巧	126
5.3	反射	126
5.3.1	反射机制的定义	126
5.3.2	反射机制的功能	126
5.3.3	利用反射机制能获得的信息	126
5.3.4	反射机制的优缺点	127
5.3.5	反射机制的应用实例	127
5.4	接口	130
5.4.1	什么是接口	130
5.4.2	接口的特征及用法	130
5.4.3	接口与抽象类的区别	131
5.5	对象克隆	131
5.5.1	为什么要克隆	131
5.5.2	如何实现克隆	132

5.5.3	浅克隆和深克隆	132
5.5.4	解决多层克隆问题	133
5.6	内部类	135
5.6.1	什么是内部类	135
5.6.2	内部类的共性	135
5.6.3	为什么需要内部类	135
5.6.4	成员内部类	137
5.6.5	局部内部类	139
5.6.6	静态内部类(嵌套类)	140
5.6.7	匿名内部类	142
6	Java 用户界面控件	143
6.1	布局	143
6.1.1	BorderLayout	143
6.1.2	FlowLayout	144
6.1.3	CardLayout	144
6.1.4	GridLayout	145
6.1.5	BoxLayout	146
6.1.6	GridBagLayout	147
6.1.7	TableLayout	149
6.2	文本框	153
6.3	复选框	156
6.4	单选钮	159
6.5	列表	163
6.5.1	构造 model	163
6.5.2	初始化 JList	164
6.5.3	添加删除元素	164
6.5.4	定制单元格渲染器	164
6.5.5	选择事件	165
6.5.6	数据事件	165
6.6	面板	166
6.6.1	Jpanel	166
6.6.2	JScrollPane	167
6.7	菜单	169
6.8	鼠标事件	175

6.8.1	MouseListener 接口	176
6.8.2	MouseMotionListener 接口	176
6.9	键盘事件	176
7	流与文件操作	178
7.1	文件和流的概念	178
7.1.1	操作系统中的文件和目录概念	178
7.1.2	流的概念	181
7.1.3	Java 的输入/输出流与文件操作概述	182
7.2	字节输入/输出流类	183
7.2.1	抽象字节流	183
7.2.2	Java 标准输入/输出	185
7.2.3	文件字节流	188
7.2.4	数据字节流	192
7.2.5	对象字节流	196
7.2.6	管道字节流	202
7.3	字符输入/输出流类	208
7.3.1	抽象字符流	208
7.3.2	文件字符流	210
7.3.3	缓冲字符流	211
7.4	文件操作	213
7.4.1	文件操作类	213
7.4.2	文件过滤器接口	215
7.4.3	随机存取文件类	216
8	Java 异常和调试	218
8.1	处理错误	218
8.2	捕获异常	220
8.2.1	try 语句块	220
8.2.2	catch 块	220
8.2.3	多个 catch 块	220
8.2.4	try 语句的嵌套	221
8.2.5	finally 语句块	221
8.2.6	throw 语句的使用	222
8.2.7	声明抛弃异常	222
8.2.8	异常丢失	224

8.3 日志	226
8.4 断言	229
8.4.1 语法形式	229
8.4.2 使用示例	230
8.4.3 注意事项	231
8.4.4 其他选项	232
8.5 调试	233
参考文献	237

1 Java 介绍

Java 是 Sun Microsystem 公司研制的一种新型的程序设计语言,它的简单、面向对象、与平台无关等优良特点,使得它在高级语言已经非常丰富的背景下脱颖而出,成为计算机和非计算机专业程序设计语言的首选。

本章将介绍 Java 的诞生和发展,Java 语言的特点及其发展历史。

1.1 了解 Java

1.1.1 Java 的诞生和发展

1.1.1.1 诞生

Java 的名字的来源:Java 是印度尼西亚爪哇岛的英文名称,因盛产咖啡而闻名。Java 语言中的许多库类名称,多与咖啡有关,如 JavaBeans(咖啡豆)、NetBeans(网络豆)以及 ObjectBeans(对象豆)等。Sun 和 Java 的标识也正如一杯正冒着热气的咖啡^[1]。

据 James Gosling 回忆,最初这个为 TV 机顶盒所设计的语言在 Sun 公司内部一直称为 Green 项目。我们的新语言需要一个名字。Gosling 注意到自己办公室外一棵茂密的橡树(Oak),这是一种在硅谷很常见的树。所以他将这个新语言命名为 Oak。但 Oak 是另外一个注册公司的名字。这个名字不可能再用了。

在命名征集会上,大家提出了很多名字。最后按大家的评选次序,将十几个名字排列成表,上报给商标律师。排在第一位的是 Silk。尽管大家都喜欢这个名字,但遭到 James Gosling 的坚决反对。排在第二和第三的都没有通过律师这一关。只有排在第四位的名字得到了所有人的认可和律师的通过,这个名字就是 Java。

20 多年来,Java 就像爪哇咖啡一样誉满全球,成为实至名归的企业级应用

平台的霸主。而 Java 语言也如同咖啡一般香醇动人。

Sun 公司的 Java 语言开发小组成立于 1991 年,其目的是开拓消费类电子产品市场,例如:交互式电视、烤面包箱等。Sun 公司内部人员把这个项目称为 Green,Sun 公司的一个叫作 Patrick Naughton 的工程师被自己开发的 C 和 C 语言编译器搞得焦头烂额,因为其中的 API 极其难用。Patrick Naughton 决定改用 NeXT,同时他也获得了研究公司的一个叫作“Stealth 计划”的项目的机会^[2-3]。

“Stealth 计划”后来改名为“Green 计划”,James Gosling 和 Mike Sheridan 也加入了 Patrick Naughton 的工作小组。该小组的领导人是 James Gosling,41 岁,于 1984 年加盟 Sun Microsystem 公司,以前在 IBM 工作。他是 Sun NeWs 窗口系统的总设计师,虽然当时 NeWs 在技术上非常成功,但在商业上却完全失败,没有 Java 那么好的运气。Gosling 是一位非常杰出的程序员,GOSMACS 也是 Gosling 写的,它是第一个用 C 实现的 EMACS 的文本编辑器。在研究开发过程中,Gosling 深刻体会到消费类电子产品和 workstation 产品的开发哲学之间的差异:消费类电子产品要求高可靠、费用低、标准化、使用简单,而 workstation 用户要求强大的计算能力,而不在乎价格以及操作的复杂性。消费类电子产品用户并不关心 CPU 的型号,也不欣赏专用昂贵的 RISC 处理器,他们需要建立在一个标准基础之上,有一系列可选的方案。从 C 开始为了使整个系统与平台无关,Gosling 首先从改写 C 编译器着手。工作小组使用的是内嵌类型平台,可以用的资源极其有限。很多成员发现 C 太复杂,以至于很多开发者经常错误使用。他们发现 C 缺少垃圾回收系统,还有可移植的安全性、分布程序设计和多线程功能。最后,他们想要一种易于移植到各种设备上的语言^[4]。

根据可用的资金,比尔·乔伊决定开发一种集 C 语言和 Mesa 语言搭成的新语言,在一份报告上,乔伊把它叫作“未来”,他提议 Sun 公司的工程师应该在 C 的基础上,开发一种面向对象的环境。最初,Gosling 试图修改和扩展 C 的功能,但是后来他放弃了。他将要创造出一种全新的语言,被他命名为“Oak”(橡树),以他的办公室外的树来命名。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,才改名为 Java)。Gosling 在开始写 Java 时,并不局限于扩充语言机制本身,更侧重于语言所运行的软硬件环境,他要建立一个系统,这个系统运行于一个巨大的、分布的、异质的网络环境中,完成各电子设备之间的通信与协同工作。Gosling 在设计中采用了虚机器码(Virtual Machine Code)方式。即 Java 语言编译后产生的是虚机器码,虚机器码运行在一个解释器上,每一个操作系统均有一个解释器,这样一来,Java 就成了平台无关语言。这和 Gosling 设计的 Sun NeWs 窗口系统有着相同的技术,在 NeWs

中用户界面统一用 Postscript 描述,不同的显示器有不同的 Postscript 解释器,这样便保证了用户界面良好的可移植性^[5-7]。

当 Patrick Naughton 加入该小组后,整个工作进展神速。Naughton 也是 Sun 公司的技术骨干,曾经是 Open Windows 项目的负责人。经过 17 个月的奋战(就像很多开发新技术的机密的工程一样,工作小组没日没夜地工作到了 1992 年的夏天),他们能够演示新平台的一部分了,包括 Green 操作系统、Oak 程序设计语言、类库和其硬件。最初的尝试是面向一种类 PDA 设备,被命名为 Star7,这种设备有鲜艳的图形界面和被称为“Duke”的智能代理来帮助用户。1992 年 12 月 3 日,这台设备进行了展示。整个系统顺利完成。它是由一个操作系统、一种语言(Java)、一个用户界面、一个新的硬件平台、三块专用芯片构成的。而通常情况下,这样的项目在 Sun 公司要 75 个人工作三年才能完成。项目完成后,在 Sun 公司内部做了一次展示和鉴定,观众的反应是:在各方面都采用了崭新的、非常大胆的技术。许多参观者对 Java 留下了非常深刻的印象,该项目得到了 Sun 的两位领导人 Scott McNealy 和 Bill Joy 的特别关注,但 Java 的前途未卜,初战告败。同年 11 月,Green 计划被转入“FirstPerson 有限公司”(一个 Sun 公司的全资子公司),团队也被重新安排到了 Artaud。FirstPerson 团队对建造一种高度互动的设备感兴趣,当时代华纳发布了一个关于电视机顶盒的征求提议书(Request for proposal)时,FirstPerson 改变了他们的目标,作为对征求意见书的响应,提出了一个机顶盒平台的提议。但是有线电视业界觉得 FirstPerson 的平台给予了用户过多的控制权,因此 FirstPerson 的投标败给了 SGI。与 3DO 公司的另外一笔关于机顶盒的交易也没有成功,由于他们的平台不能在电视工业产生任何效益,公司再并回 Sun 公司。FirstPerson 一直到 1994 年初仍在坚持开发专用的顶置盒式操作系统,但实践证明,这个市场尚未成熟。例如:好几家交互式有线电视网的实验都以失败告终。Naughton 立即认识到研究方向应该转移。在 Naughton 的努力下,研究开始向在线服务、CD-ROM 和基于网络的台式机方向转移。最后 FirstPerson 解散,一部分人去了 Sun Interactive 开发数字视频数据服务器,而另一部分人开始把 Java 用于基于网络的台式机系统^[8]。

Java 语言的转折点是 1994 年。1994 年 6、7 月间,在经历了一场历时三天的讨论之后,团队决定再一次改变努力的目标,这次他们决定将该技术应用于万维网。他们认为随着 Mosaic 浏览器的出现,因特网正在向同样的高度互动的远景演变,而这一远景正是他们在有线电视网中看到的。作为原型,Patrick Naughton 写了一个小型万维网浏览器 WebRunner,后来改名为 HotJava。同年,Oak 改名为 Java。WWW 已如火如荼地发展起来。Gosling 意识到 WWW

需要一个中性的浏览器,它不依赖于任何硬件平台和软件平台。它应是一种实时性较高,可靠安全,有交互功能的浏览器。于是 Gosling 决定用 Java 开发一个新的 Web 浏览器。

1.1.1.2 发展历史

Java 从第一个版本诞生到现在已经有 19 年的时间了。19 年过去了,如今 JDK 已经发展到了 1.8 版。在这 19 年里还诞生了无数与 Java 相关的产品、技术和标准。现在让我们走入时间隧道,从孕育 Java 语言的时代开始,回顾一下 Java 的发展轨迹和历史变迁。

1991 年 4 月,由 James Gosling 博士领导的绿色计划(Green Project)开始启动,此计划的目的是开发一种能够在各种消费性电子产品(如机顶盒、冰箱、收音机等)上运行的程序架构。这个计划的产品就是 Java 语言的前身:Oak(橡树)。Oak 当时在消费品市场上并不算成功,但随着 1995 年互联网潮流的兴起,Oak 迅速找到了最适合自己发展的市场定位并蜕变为 Java 语言。

1992 年 3 月,由于 Oak 已被用作另一种已存在的编程语言名称,因此必须选一个新的名字——它就是 Java,灵感来源于咖啡。

1993 年 2 月,FirstPerson 试图从时代华纳获得一个电视机顶盒交互系统的一揽子订单。在那时,由于绿色计划不是很成功,随即失去了时代华纳的订单。于是开发的重心从家庭消费电子产品转到了电视机顶盒的相关平台上。

1995 年 5 月 23 日,Oak 语言改名为 Java,并且在 SunWorld 大会上正式发布 Java 1.0 版本。Java 语言第一次提出了“Write Once, Run Anywhere”的口号。

1996 年 1 月 23 日,JDK 1.0 发布,Java 语言有了第一个正式版本的运行环境。JDK 1.0 提供了一个纯解释执行的 Java 虚拟机(Sun Classic VM)。JDK 1.0 版本的代表技术包括:Java 虚拟机、Applet、AWT 等。

1996 年 4 月,10 个最主要的操作系统供应商声明将在其产品中嵌入 Java 技术。同年 9 月,已有大约 8.3 万个网页应用了 Java 技术来制作。在 1996 年 5 月底,Sun 公司于美国旧金山举行了首届 JavaOne 大会,从此 JavaOne 成为全世界数百万 Java 语言开发者每年一度的技术盛会。

1997 年 2 月 19 日,Sun 公司发布了 JDK 1.1,Java 技术的一些最基础的支撑点(如 JDBC 等)都是在 JDK 1.1 版本中发布的,JDK 1.1 版的代表技术有:JAR 文件格式、JDBC、JavaBeans、RMI。Java 语法也有了一定的发展,如内部类(Inner Class)和反射(Reflection)都是在这个时候出现的。

截至 1999 年 4 月 8 日,JDK 1.1 一共发布了 JDK 1.1.0~JDK 1.1.8 九个版本。从 JDK 1.1.4 之后,每个 JDK 版本都有一个自己的名字(工程代号),分

别为:JDK 1.1.4 Sparkler(宝石)、JDK 1.1.5 Pumpkin(南瓜)、JDK 1.1.6 Abigail(阿比盖尔,女子名)、JDK 1.1.7 Brutus(布鲁图,古罗马政治家和将军)和 JDK 1.1.8 Chelsea(切尔西,城市名)^[9-10]。

1998年12月4日,JDK迎来了一个里程碑式的版本JDK 1.2,工程代号为Playground(竞技场),Sun在这个版本中把Java技术体系拆分为3个方向,分别是面向桌面应用开发的J2SE(Java 2 Platform, Standard Edition)、面向企业级开发的J2EE(Java 2 Platform, Enterprise Edition)和面向手机等移动终端开发的J2ME(Java 2 Platform, Micro Edition)。在这个版本中出现的代表性技术非常多,如EJB、Java Plug-in、Java IDL、Swing等,并且这个版本中Java虚拟机第一次内置了JIT(Just In Time)编译器(JDK 1.2中曾并存过3个虚拟机,Classic VM、HotSpot VM和Exact VM,其中Exact VM只在Solaris平台出现过;后面两个虚拟机都是内置JIT编译器的,而之前版本所带的Classic VM只能以外挂的形式使用JIT编译器)。在语言和API级别上,Java添加了strictfp关键字与现在Java编码之中常用的一系列Collections集合类。

在1999年3月和7月,分别有JDK 1.2.1和JDK 1.2.2两个小版本发布。

1999年4月27日,HotSpot虚拟机发布,HotSpot最初由一家名为“Longview Technologies”的小公司开发,因为HotSpot的优异表现,这家公司在1997年被Sun公司收购了。HotSpot虚拟机发布时是作为JDK 1.2的附加程序提供的,后来它成为JDK 1.3及之后所有版本的Sun JDK的默认虚拟机。

2000年5月8日,工程代号为Kestrel(美洲红隼)的JDK 1.3发布,JDK 1.3相对于JDK 1.2的改进主要表现在一些类库上(如数学运算和新的Timer API等),JNDI服务从JDK 1.3开始被作为一项平台级服务提供(以前JNDI仅仅是一项扩展),使用CORBA IIOP来实现RMI的通信协议,等等。这个版本还对Java 2D做了很多改进,提供了大量新的Java 2D API,并且新添加了JavaSound类库。JDK 1.3有1个修正版本JDK 1.3.1,工程代号为Ladybird(瓢虫),于2001年5月17日发布。

自从JDK 1.3开始,Sun维持了一个习惯:大约每隔两年发布一个JDK的主版本,以动物命名,期间发布的各个修正版本则以昆虫作为工程名称。

2002年2月13日,JDK 1.4发布,工程代号为Merlin(灰背隼)。JDK 1.4是Java真正走向成熟的一个版本,Compaq、Fujitsu、SAS、Symbian、IBM等著名公司都有参与甚至实现自己独立的JDK 1.4。哪怕是在十多年后的今天,仍然有许多主流应用(Spring、Hibernate、Struts等)能直接运行在JDK 1.4之上,或者继续发布能运行在JDK 1.4上的版本。JDK 1.4同样发布了很多新的技术,如正则表达式、异常链、NIO、日志类、XML解析器和XSLT转换器等。

JDK 1.4 有两个后续修正版,分别为 2002 年 9 月 16 日发布的工程代号为 Grasshopper(蚱蜢)的 JDK 1.4.1 和 2003 年 6 月 26 日发布的工程代号为 Mantis(螳螂)的 JDK 1.4.2。

2002 年前后还发生了一件与 Java 没有直接关系,但事实上对 Java 的发展进程影响很大的事件,那就是微软公司的 .NET Framework 发布了。这个无论是技术实现上还是目标用户上都与 Java 有很多相近之处的技术平台给 Java 带来了许多讨论、比较和竞争,.NET 平台和 Java 平台之间声势浩大的孰优孰劣的论战到目前为止都在继续。

2004 年 9 月 30 日,JDK 1.5 发布,工程代号 Tiger(老虎)。从 JDK 1.2 以来,Java 在语法层面上的变化一直很小,而 JDK 1.5 在 Java 语法易用性上做出了非常大的改进。例如,自动装箱、泛型、动态注解、枚举、可变长参数、遍历循环(foreach 循环)等语法特性都是在 JDK 1.5 中加入的。在虚拟机和 API 层面上,这个版本改进了 Java 的内存模型(Java Memory Model,JMM),提供了 java.util.concurrent 并发包等。另外,JDK 1.5 是官方声明可以支持 Windows 9x 平台的最后一个 JDK 版本。

2006 年 12 月 11 日,JDK 1.6 发布,工程代号 Mustang(野马)。在这个版本中,Sun 终结了从 JDK 1.2 开始已经有 8 年历史的 J2EE、J2SE、J2ME 的命名方式,启用 Java SE 6、Java EE 6、Java ME 6 的命名方式。JDK 1.6 的改进包括:提供动态语言支持(通过内置 Mozilla Java Rhino 引擎实现)、提供编译 API 和微型 HTTP 服务器 API 等。同时,这个版本对 Java 虚拟机内部做了大量改进,包括锁与同步、垃圾收集、类加载等方面的算法都有相当多的改动。

在 2006 年 11 月 13 日的 JavaOne 大会上,Sun 公司宣布最终会将 Java 开源,并在随后的一年多时间内,陆续将 JDK 的各个部分在 GPL v2(GNU General Public License v2)协议下公开了源码,并建立了 OpenJDK 组织对这些源码进行独立管理。除了极少量的产权代码(Encumbered Code,这部分代码大多是 Sun 本身也无权限进行开源处理的)外,OpenJDK 几乎包括了 Sun JDK 的全部代码,OpenJDK 的质量主管曾经表示,在 JDK 1.7 中,Sun JDK 和 OpenJDK 除了代码文件头的版权注释之外,代码基本上完全一样,所以 OpenJDK 7 与 Sun JDK 1.7 本质上就是同一套代码库开发的产品。

JDK 1.6 发布以后,由于代码复杂性的增加,JDK 开源,开发 JavaFX,经济危机及 Sun 收购案等原因,Sun 在 JDK 发展以外的事情上耗费了很多资源,JDK 的更新没有再维持两年发布一个主版本的发展速度。JDK 1.6 到目前为止一共发布了 37 个 Update 版本,最新的版本为 Java SE 6 Update 37,于 2012 年 10 月 16 日发布。