

普通高等教育“计算机类专业”规划教材

Java程序设计 与项目实训教程（第2版）

张志锋 邓璐娟 张建伟 宋胜利 等编著



清华大学出版社

普通高等教育“计算机类专业”规划教材

Java程序设计 与项目实训教程

(第2版)

张志锋 邓璐娟 张建伟 宋胜利



等编著

清华大学出版社

北京

内 容 简 介

本书旨在培养学生的 Java 工程实践能力和计算机系统能力。

全书理论联系实践,基于以项目为驱动的教学模式,详细、系统化讲解 Java 技术。全书共 12 章,主要内容包括 Java 语言的基础知识、Java 的常用开发工具、Java 语言的基础语法、Java 核心技术、Java 语言的异常处理、Java 语言的图形用户界面组件、数据库编程技术、流与文件、多线程以及网络编程技术。本书以项目实践贯穿知识体系。通过 2 个实践项目的练习,使学生能够在掌握基本理论知识的同时,培养和提高综合应用实践能力。

本书可作为普通高等院校的 Java 程序设计相关课程教材,也可作为 Java 工程实践教材以及 Java 软件开发人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计与项目实训教程/张志锋等编著. —2 版. —北京:清华大学出版社,2017

(普通高等教育“计算机类专业”规划教材)

ISBN 978-7-302-47311-4

I. ①J… II. ①张… 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(2017)第 124420 号

责任编辑:白立军

封面设计:常雪影

责任校对:白 雷

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:保定市中华美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:29.75

字 数:685 千字

版 次:2012 年 1 月第 1 版 2017 年 8 月第 2 版

印 次:2017 年 8 月第 1 次印刷

印 数:1~2000

定 价:59.00 元

产品编号:070256-01

前 言

本书的编写宗旨是：切实贯彻、实践“工程教育认证”的理念，加强学生的专业应用能力、系统设计能力和工程实践能力培养。本教材引进以项目为驱动的教学模式，通过 2 个完整的项目和 127 个应用示例的实践训练，使读者在学习 Java 技术知识体系的同时，结合工程实践加深理解和巩固所学知识，同时培养学生的工程意识、协作精神以及综合应用所学知识解决实际问题的能力。

作者编写的 Java 方向系列教材与本教材具有同样的风格，均基于以项目为驱动的教学模式，属于同系列的教材。

本书主要章节以及具体安排如下。

第 1 章介绍 Java 语言的基础知识。通过本章学习，读者能够对 Java 技术有初步的了解。

第 2 章介绍 Java 常用开发工具，如 JDK、NetBeans、Eclipse。

第 3 章介绍 Java 语言的基本语法。

第 4 章介绍 Java 技术核心内容。本章是面向对象程序设计的基础。

第 5 章介绍 Java 语言的异常处理机制。

第 6 章介绍 Java 语言常用的图形用户界面组件，为第 8 章和第 12 章的项目实训奠定基础。

第 7 章介绍使用 JDBC 技术连接 MySQL、SQL Server 数据库，使学生掌握数据库编程基本技能，为后面的项目实训奠定基础。

第 8 章通过一个项目综合练习前 7 章所学知识。可以在讲解前几章之前先介绍第 8 章的项目，也可以把第 8 章的内容分解到前 7 章中，结合该项目实现的各个环节讲解前面各章的知识点。

第 9 章介绍 Java 语言中流与文件的使用。

第 10 章介绍 Java 语言的多线程相关技术。

第 11 章介绍 Java 语言的网络编程相关技术。

第 12 章是一个综合项目开发实例。在第 8 章项目实训的基础上，整合前 11 章的内容，实现一个多线程的 C/S 模式网络应用程序（网络聊天系统）。

本书由张志锋、邓璐娟任主编，张建伟、宋胜利、孙玉胜、武丰龙任副主编。参与本书编写的有马军霞、梁树军、郑倩、马欢和连清平。

在本书的编写和出版过程中得到了郑州轻工业学院、清华大学出版社的大力支持和帮助，在此表示感谢。

由于编写时间仓促，水平有限，书中难免有疏漏之处，敬请读者不吝赐教。

除了配套制作的教学课件、教学日历、教学大纲、实验大纲、期末试卷外，本书还提供书中示例的源代码、课后习题参考答案、电子版课后习题以及其他未收入教材的 Java 实训项目（可在清华大学出版社网站下载：www.tup.com.cn），并为教师提供服务 QQ 和邮箱（2394115659.qq.com），以提供更多更便捷的教学资源服务。

编 者

2017 年 1 月

· I ·

目 录

第 1 章 Java 语言概述	1
1.1 计算机语言的发展	1
1.2 Java 语言简介	3
1.2.1 Java 语言的发展	4
1.2.2 Java 语言的特点	5
1.2.3 Java 程序的工作原理	7
1.3 Java 应用程序	8
1.3.1 Application 应用程序	8
1.3.2 Applet 小应用程序	12
1.4 一切皆为对象	14
1.5 常见问题及解决方案	14
1.6 本章小结	19
1.7 习题	20
第 2 章 Java 语言开发环境	22
2.1 JDK 安装配置	22
2.1.1 JDK 简介与下载	22
2.1.2 JDK 安装与配置	22
2.2 NetBeans 开发环境	26
2.2.1 NetBeans 简介与下载	26
2.2.2 NetBeans 的安装与使用	27
2.3 Eclipse 开发环境	33
2.3.1 Eclipse 简介与下载	33
2.3.2 Eclipse 的使用	34
2.4 常见问题及解决方案	39
2.5 小结	39
2.6 习题	39
第 3 章 Java 语言基础	40
3.1 Java 语言的基本语法	40
3.1.1 标识符	40
3.1.2 关键字	41
3.1.3 分隔符	43
3.1.4 数据类型	44

3.1.5	常量和变量	47
3.1.6	运算符与表达式	49
3.2	控制语句	53
3.2.1	顺序语句	54
3.2.2	选择语句	55
3.2.3	循环语句	59
3.2.4	转移语句	65
3.3	数组	67
3.3.1	一维数组	67
3.3.2	二维数组	72
3.4	字符串	74
3.4.1	声明字符串变量	74
3.4.2	字符串的运算	75
3.4.3	String 类的常用方法	76
3.5	常见问题及解决方案	77
3.6	本章小结	79
3.7	习题	79
第 4 章	Java 语言面向对象程序设计	82
4.1	面向对象的概念	82
4.1.1	面向对象程序设计	82
4.1.2	面向对象程序设计的术语	83
4.1.3	面向对象程序设计的特性	83
4.1.4	面向对象程序设计的优点	83
4.2	类和对象	84
4.2.1	类	84
4.2.2	对象	87
4.3	类的封装性	88
4.3.1	构造方法	89
4.3.2	成员方法	91
4.3.3	访问权限	91
4.3.4	this、static、final 和 instanceof	92
4.4	类的继承性	96
4.4.1	父类和子类	96
4.4.2	子类的声明与方法的覆盖	97
4.4.3	super	98
4.4.4	类的封装性和继承性的程序应用	99
4.5	类的多态性	108
4.5.1	多态性的概念	108

4.5.2	方法的重载和覆盖	109
4.5.3	多态性程序应用	109
4.6	包	114
4.6.1	包的概念	114
4.6.2	包的创建和包对文件的管理	115
4.6.3	包的导入	116
4.7	抽象类与接口	116
4.7.1	抽象类	116
4.7.2	接口	117
4.8	内部类与匿名类	124
4.8.1	内部类	124
4.8.2	匿名类	125
4.9	常见问题及解决方案	125
4.10	本章小结	127
4.11	习题	128
第5章	异常处理	130
5.1	Java 异常处理的基本概念	130
5.1.1	错误与异常	130
5.1.2	错误和异常的分类	131
5.2	异常处理	133
5.2.1	捕获异常并处理	133
5.2.2	抛出异常	134
5.3	自定义异常类	137
5.4	常见问题及解决方案	138
5.5	本章小结	139
5.6	习题	139
第6章	图形用户界面	141
6.1	Swing 简介	141
6.2	Swing 的组件	141
6.2.1	Swing 组件关系	142
6.2.2	JFrame 和 JLabel	143
6.2.3	JDialog 和 JOptionPane	146
6.2.4	JTextField 和 JPasswordField	148
6.2.5	JButton、JCheckBox 和 JRadioButton	150
6.2.6	JComboBox、JList、JTextArea 和 JScrollPane	151
6.2.7	JPanel 和 JSlider	152
6.3	布局管理器	153

6.3.1	布局管理器的概念	154
6.3.2	FlowLayout	154
6.3.3	BorderLayout	155
6.3.4	GridLayout	156
6.3.5	BoxLayout	157
6.3.6	CardLayout	158
6.3.7	GroupLayout	159
6.4	Java 中的事件处理	162
6.4.1	事件处理的基本概念	162
6.4.2	事件和事件源	164
6.4.3	注册监听器	165
6.4.4	事件处理	167
6.4.5	鼠标事件处理	169
6.4.6	键盘事件处理	171
6.5	图形用户界面的高级组件	173
6.5.1	菜单	173
6.5.2	表格	178
6.5.3	JTree	185
6.6	常见问题及解决方案	202
6.7	本章小结	203
6.8	习题	204
第 7 章	数据库编程	206
7.1	JDBC 介绍	206
7.1.1	什么是 JDBC	206
7.1.2	JDBC 的结构	207
7.2	通过 JDBC 驱动访问数据库	207
7.2.1	通过 JDBC 访问 MySQL 数据库	207
7.2.2	通过 JDBC 访问 Microsoft SQL Server 数据库	212
7.3	查询数据库	218
7.4	更新数据库(增、删、改)	224
7.5	学生信息管理系统项目实训	225
7.6	常见问题及解决方案	238
7.7	本章小结	239
7.8	习题	239
第 8 章	资费管理系统项目实训	241
8.1	项目需求说明	241
8.2	项目分析与设计	242

8.3	项目的数据库设计	242
8.4	项目实施	244
8.4.1	项目的模块划分及其结构	244
8.4.2	项目的登录和注册功能设计与实现	244
8.4.3	项目主界面设计与实现	253
8.4.4	项目的用户管理功能设计与实现	256
8.4.5	项目资费管理功能设计与实现	267
8.4.6	项目其他功能模块的设计与实现	277
8.5	常见问题及解决方案	277
8.6	本章小结	277
8.7	习题	277
第9章	I/O 流与文件	278
9.1	文件与流简介	278
9.1.1	文件简介	278
9.1.2	流简介	279
9.2	字节输入输出流	280
9.2.1	InputStream 和 FileInputStream	280
9.2.2	OutputStream 和 FileOutputStream	282
9.2.3	DataInputStream 和 DataOutputStream	286
9.2.4	ObjectInputStream 和 ObjectOutputStream	288
9.2.5	BufferedInputStream 和 BufferedOutputStream	291
9.2.6	标准的输入输出流	293
9.3	字符输入输出流	296
9.3.1	Reader 和 Writer	296
9.3.2	FileReader 和 FileWriter	297
9.3.3	BufferedReader 和 BufferedWriter	298
9.4	文件操作类	301
9.4.1	文件类	301
9.4.2	随机访问文件类	303
9.4.3	文件过滤器接口	305
9.4.4	文件对话框类	307
9.5	常见问题及解决方案	308
9.6	本章小结	309
9.7	习题	309
第10章	多线程	311
10.1	多线程的概念	311
10.1.1	程序、进程和线程	311

10.1.2	使用线程的好处	312
10.2	线程的实现	313
10.2.1	继承 Thread 线程类	313
10.2.2	实现 Runnable 接口	316
10.2.3	使用 Timer 类和继承 TimerTask 类	317
10.3	线程的生命周期	318
10.3.1	线程的状态	318
10.3.2	线程的优先级	319
10.3.3	线程的调度	320
10.4	线程的同步	326
10.4.1	线程间的关系	326
10.4.2	线程同步问题	328
10.5	常见问题及解决方案	331
10.6	本章小结	332
10.7	习题	332
第 11 章	网络编程	334
11.1	网络通信概念	334
11.2	统一资源定位器(URL)的使用	335
11.3	Java 网络编程	339
11.3.1	Java 网络编程概述	339
11.3.2	基于 TCP 的 Socket 编程原理	341
11.3.3	基于 TCP 的 Socket 编程实现	342
11.3.4	基于 UDP 的 Socket 编程原理	352
11.3.5	基于 UDP 的 Socket 编程实现	353
11.3.6	基于 SSL 的 Socket 编程原理	356
11.4	常见问题及解决方案	357
11.5	本章小结	357
11.6	习题	358
第 12 章	网络聊天系统项目实训	360
12.1	C/S 模式	360
12.2	项目需求分析	360
12.3	项目设计	361
12.3.1	服务器端设计	361
12.3.2	客户端设计	362
12.3.3	通信协议设计	362
12.4	项目的数据库设计	363
12.5	项目的开发过程	364

12.5.1	项目简介.....	364
12.5.2	网络通信系统服务器端实现.....	366
12.5.3	聊天系统客户端实现.....	405
12.5.4	聊天系统功能演示.....	451
12.6	常见问题及解决方案.....	461
12.7	本章小结.....	461
12.8	习题.....	461
参考文献.....		462

第 1 章 Java 语言概述

Java 语言作为一种优秀的面向对象语言,具有简单、稳定、可移植、多线程和网络安全等优良特性,已经成为目前软件开发首选的面向对象语言。Java 语言不仅可以开发大型的商业应用软件,也可以开发应用于 Java Web 网站的应用软件。本章主要讲解 Java 语言的相关概念、原理和简单程序设计。

本章主要内容:

- 计算机语言的发展。
- Java 语言的发展。
- Java 语言的特点。
- Java 语言的工作原理。
- Java 应用程序。
- 程序中常见问题及解决方案。

1.1 计算机语言的发展

计算机的运行离不开软件,软件是计算机的灵魂。软件由一系列程序和相关的数据组成。用来编写程序的技术称为计算机语言,又称为程序设计语言。随着计算机技术和操作系统的发展,不同风格的程序设计语言不断出现。计算机语言经历了由低级语言到高级语言的发展过程。按其是否接近人类自然语言,可将计算机语言划分为三大类:机器语言、汇编语言和高级语言。

1. 机器语言

软件的产生始于早期的机械式计算机的开发。从 19 世纪起,随着机械式计算机的更新,出现了穿孔卡片,这种卡片可以指导计算机进行工作。

但是直到 20 世纪中期现代化的电子计算机出现之后,软件才真正得以飞速发展。1946 年,第一台计算机(ENIAC)在美国宾夕法尼亚州诞生,在 ENIAC 上使用的也是穿孔卡片,在卡片上使用的是专家们才能理解的语言,由于它与人类语言的差别极大,所以称之为机器语言。机器语言也就是第一代计算机语言。

这种机器语言是最原始的计算机语言,是直接用二进制代码指令表达的计算机语言。指令是用 0 和 1 组成的一串代码,它们有一定的位数,并分成若干段,各段的编码表示不同的含义。例如,某台计算机字长为 16 位,即由 16 个二进制数组成一条指令或其他信息。16 个 0 和 1 可组成各种排列组合,通过线路变成电信号,让计算机执行各种不同的操作。如某类型计算机的指令 1011011000000000 表示让计算机进行一次加法操作,而指令 1011010100000000 则表示进行一次减法操作。它们的前 8 位表示操作码,而后 8 位表示地址码。从上面两条指令可以看出,它们的差别只是操作码中从左边第 0 位算起的第 7 和第 8 位不同。

机器语言是计算机唯一可直接识别的语言,或者说用机器语言编写的程序可以在计算机上直接执行。用机器语言编写程序是十分困难的,也容易出错,不易修改,程序可读性极差。另外,由于不同类型的计算机具有不同的指令系统,在某一类计算机上编写的程序不能够在另一类计算机上运行,可移植性差。这种语言本质上是计算机能识别的唯一语言,但人类却很难理解它。以后的计算机语言就是在这个基础上,将机器语言越来越简化到人类能够直接理解的、近似于人类语言的程度,但最终送入计算机的工作语言还是这种机器语言。高级语言的任务就是将它翻译成人类易懂的语言,而这个翻译工作可以由计算速度越来越高、工作越来越可靠的计算机自己来完成。

2. 汇编语言

计算机语言发展到第二代,出现了汇编语言。汇编语言是一种符号语言,使用一些容易记忆的助记符来代替机器指令。用汇编语言编写的程序相对于机器语言来说可读性好,容易编程,修改也方便。但是计算机不能够直接执行用汇编语言编写的程序。汇编语言源程序必须通过语言处理程序将其翻译成对应的机器语言,才能被计算机识别、执行。汇编语言和机器语言没有本质的差别,基本上一条语句对应着一条指令。用汇编语言编程最主要的缺点是程序与所要解决问题的数学模型之间的关系不直观,编程难度较大。和机器语言一样,汇编语言程序的可移植性也差。

比起机器语言,汇编语言大大前进了一步,尽管它还是太复杂,人们在使用时很容易出错误,但毕竟许多编码已经开始用字母来代替。简单的0、1数码谁也不会理解,但字母是人们能够阅读并拼写的。第二代计算机语言仍然是“面向机器”的语言,但它已注定要成为机器语言向更高级语言进化的桥梁。

一般把机器语言和汇编语言称为低级语言。

3. 高级语言

当计算机语言发展到第三代时,就进入了“面向人类”的语言阶段。在最初与计算机交流的过程中,人们意识到,应该设计一种语言,该语言接近于人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用,这就是高级语言。

高级语言又称为算法语言,它是独立于机型、面向应用、实现算法的语言。高级语言从根本上摆脱了指令系统的束缚,语言描述接近于人类语言,人们不必熟悉计算机具体的内部结构和指令,只需把精力集中在问题的描述和求解上。

FORTRAN语言是世界上第一个被正式推广使用的高级语言。它是1954年被提出来的,1956年开始正式使用,至今已有60多年的历史,仍历久不衰,始终是数值计算领域所使用的主要语言。

几十年来,共有2600多种高级语言出现,其中具有代表性的语言如下。

1954年FORTRAN语言诞生,1958年ALGOL语言诞生,1960年LISP和COBOL语言诞生,1962年APL和SIMULA语言诞生,1964年BASIC和PL/I语言诞生,1966年ISWIM语言诞生,1967年Simulator语言诞生,1970年Prolog语言诞生,1972年C语言诞生,1975年Pascal和Scheme语言诞生,1977年OPS5语言诞生,1978年CSP和FP语言诞生,1980年dBASE II语言诞生,1983年Smalltalk-80、Ada和Parlog语言诞生,1984年Standard ML语言诞生,1986年C++、CLP(R)和Eiffel语言诞生,1987年Perl语言诞生,1988年CLOS,Mathematica和Oberon语言诞生,1990年Haskell语言诞生,1991年Python语

言诞生,1995 年 Java、PHP 和 Ruby 语言诞生,2002 年 C# 语言诞生。

高级语言程序设计思想又经历了面向问题、面向过程和面向对象的发展过程。随着 Windows 操作系统的普及,又出现了面向对象的可视化编程语言,比较流行的有 Visual Basic、Visual C++、Visual C# 和 Java 等。

20 世纪 60 年代中后期,软件越来越多,规模越来越大,而软件的开发基本上是各自为战,缺乏科学规范的系统规划与测试、评估标准,其恶果是大批耗费巨资建立起来的软件系统,由于含有错误而无法使用,甚至带来巨大损失,软件给人的感觉是越来越不可靠,以致几乎没有不出错的软件。这一切,极大地震动了计算机界,史称“软件危机”。人们认识到,大型程序的编制不同于编写小程序,它应该是一项新的技术,应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性,也便于验证正确性。1969 年,提出了结构化程序设计方法,1970 年,第一个结构化程序设计语言——Pascal 语言出现,标志着结构化程序设计时期的开始。

20 世纪 80 年代初期开始,在软件设计思想上又产生了一次革命,其成果就是面向对象的程序设计。在此之前的高级语言几乎都是面向过程的,程序的执行是流水线似的,在一个模块被执行完成前,人们不能干别的事,也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的,对人而言是希望发生一件事就处理一件事,也就是说,不能面向过程,而应是面向具体的应用功能,也就是对象(object)。其方法就是软件的集成化,如同硬件的集成电路一样,生产一些通用的、封装紧密的功能模块,称之为软件集成块或模块,它与具体应用无关,但能相互组合,完成具体的应用功能,同时又能重复使用。对使用者来说,只关心它的接口及能实现的功能,至于它是如何实现的,那是它内部的事,使用者完全不关心,C++、Visual Basic、Delphi、C# 和 Java 就是典型代表。

下一代语言(又称为第四代)是使用第二代和第三代语言编制而成的,每一种语言都有其特定的应用范围。实际上,语言发展到今天已出现了一些有实用性的第四代语言。第四代语言的特点就是它们只需要操作人员输入原始数据,并命令它们执行。至于怎样执行则由语言本身来决定,它已经在相当程度上替代了人脑的工作。第四代语言的特点还在于:操作者几乎不需要经过特殊训练,几乎所有的“实用语言”都有“帮助”功能,可以遵照计算机给出的指示来完成工作。

高级语言的下一个发展目标是面向应用,只需要告诉程序要干什么,程序就能自动生成算法,自动进行处理,这就是非过程化的程序语言。

计算机语言的未来发展趋势:面向对象程序设计以及数据抽象在现代程序设计思想中占有很重要的地位,未来语言的发展将不再是一种单纯的语言标准,将会完全面向对象,更易于表达现实世界,更便于编写,其使用者将不再只是专业的编程人员,人们完全可以以制定真实生活中一项工作流程的简单方式来完成编程。

计算机技术的飞速发展,离不开人类科技知识的积累,离不开许许多多热衷于此并呕心沥血的科学家们的探索。正是一代代技术的积累才构筑了今天的信息化成就。

1.2 Java 语言简介

Java 语言是由原 Sun 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称。

Java 平台由 Java 虚拟机 (Java Virtual Machine, JVM) 和 Java 应用编程接口 (Application Programming Interface, API) 构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分。在硬件或操作系统平台上安装一个 Java 平台之后,Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序只需编译一次,就可以在各种系统上运行。

Java 分为 3 种平台: Java SE (Java Platform Standard Edition, Java 标准版平台)、Java EE (Java Platform Enterprise Edition, Java 企业版平台) 和 Java ME (Java Platform Micro Edition, Java 微型版平台)。

1. Java SE

Java SE 以前称为 J2SE。主要用于开发和部署在桌面和服务端端的 Java 应用程序。Java SE 包含了支持 Java Web 服务开发的类,并为 Java EE 提供基础。本书使用的是 Java SE。

2. Java EE

Java EE 以前称为 J2EE。主要用于开发和部署企业版可移植的、健壮的、可伸缩的且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的,它提供 Web 服务、组件模型、管理和通信 API,可以用来实现企业级的面向服务体系结构 (Service-Oriented Architecture, SOA) 和 Web 2.0 应用程序。

3. Java ME

Java ME 以前称为 J2ME。主要用于开发和部署移动设备和嵌入式设备 (如手机、PDA、电视机顶盒和打印机) 的 Java 应用程序。

1.2.1 Java 语言的发展

Java 语言起源于 1991 年,是原 Sun 公司为一些智能消费性电子产品设计的一个通用语言 (Oak)。Oak 语言是 Java 语言的前身。当时, Oak 并没有引起人们的注意,项目最初的目的只是为了开发一种独立于平台的软件技术,而且在网络出现之前, Oak 可以说是默默无闻,甚至差点夭折。但是,网络的出现改变了 Oak 的命运。

在 Java 出现以前, Internet 上的信息内容都是一些静态的 HTML 文档。这对于那些迷恋于 Web 浏览的人们来说简直不可容忍。他们迫切希望能在 Web 中看到一些交互式的内容,开发人员也很希望能够在 Web 上创建无须考虑软硬件平台就可以执行的应用程序,当然这些程序还要有极大的安全保障。对于用户的这种要求,传统的编程语言显得无能为力。1994 年,随着互联网和 Web 技术的飞速发展, Sun 公司用 Java 编制了 HotJava 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,得以研发和发展。为了促销和法律的原因, 1995 年 Oak 更名为 Java。很快 Java 被工业界认可,许多大公司如 IBM、Microsoft、DEC 等购买了 Java 的使用权,并被美国 PC Magazine 评为 1995 年十大优秀科技产品。

Java 发展历程中的重大事件如下。

1995 年 5 月 23 日, Java 语言诞生。

1996 年 1 月, 第一个 Java 语言开发环境 JDK 1.0 诞生。

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。

1996 年 9 月, 约 8.3 万个网页应用了 Java 技术来制作。

1997 年 2 月 18 日, JDK 1.1 发布。

1997 年 4 月 2 日, JavaOne 会议召开, 参与者逾万人, 创当时全球同类会议规模纪录。

1997 年 9 月, JavaDeveloperConnection 社区成员超过十万。

1998 年 2 月, JDK 1.1 被下载超过 2 000 000 次。

1998 年 12 月 8 日, Java 2 企业平台 J2EE 发布。

1999 年 6 月, Sun 公司发布 Java 的 3 个版本: 标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。

2000 年 5 月 8 日, JDK 1.3 发布。

2000 年 5 月 29 日, JDK 1.4 发布。

2001 年 6 月 5 日, Nokia 公司宣布, 到 2003 年将出售 1 亿部支持 Java 的手机。

2001 年 9 月 24 日, J2EE 1.3 发布。

2002 年 2 月 26 日, J2SE 1.4 发布, 自此 Java 的计算能力有了大幅提升。

2004 年 9 月 30 日, J2SE 1.5 发布, 成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE 1.5 更名为 Java SE 5.0。从此, Java 的各种版本全部更名, 以取消其中的数字 2: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2005 年 6 月, JavaOne 大会召开, Sun 公司公开 Java SE 6(1.6)。

2006 年 12 月, Sun 公司发布 JRE 6.0。

2009 年 4 月 20 日, 甲骨文(Oracle)公司宣布收购 Sun 公司。该交易价值约为 74 亿美元。

2010 年 1 月 21 日, 甲骨文公司宣布正式完成对 Sun 公司的收购。昔日的“红色巨人”Sun 公司走过了 Java 发展的核心时期。“太阳落山了, 红色巨人崛起”也许是未来人们对此次收购的评价。Sun 公司的负责人说“这是一个旅途的开始”, 希望红色巨人能够崛起, 希望这场旅途不会结束。我们期待 Java 能够迎来新的发展机遇。正如甲骨文公司 CEO 拉里·埃里森(Larry Ellison)所说, “我们收购 Sun 公司将改变 IT 业, 整合第一流的企业软件和关键任务计算系统。甲骨文公司将成为业界唯一一家提供综合系统的厂商, 系统的性能、可靠性和安全性将有所提高, 而价格将会下滑。”自 2005 年以来, 甲骨文公司已经收购了 51 家公司, 如仁科、BEA、MySQL 等, Sun 公司是第 52 家。一旦被收购者的产品线一旦整合完成, 甲骨文公司就有了成为 IT 界巨人的资格。

原 Sun 公司宣布, 2010 年 9 月, JDK 7.0 发布。但由于甲骨文公司收购的原因, JDK 7.0 被推迟到 2010 年底才发布。

2014 年 3 月, 甲骨文公司发布 JDK 8。

1.2.2 Java 语言的特点

Sun 公司对 Java 的定义是: “Java: A simple, object-oriented, distributed, robust, secure, architecture-neutral, portable, high-performance, multi-threaded and dynamic language.”即 Java 是一种简单的、面向对象的、分布式的、健壮的、安全的、体系结构独立的、可移植的、高性能的、多线程的动态语言。

1. Java 语言是简单的

Java 语言是在 C 和 C++ 语言的基础上进行简化和改进的新的语言。Java 去掉了 C 和 C++ 中的不易掌握和理解的多继承、指针和内存管理等。Java 提供了自动的“垃圾”回收机

制,使程序员不必为内存管理而担忧。Java 的简单性降低了学习的难度,提高了程序的性能。

2. Java 语言是面向对象的

面向对象是 Java 语言的重要特征。Java 语言没有采用传统的以过程为中心的编程方法,而是采用以对象为中心的、模拟人类社会和人解决实际问题的方法的面向对象的程序设计语言。因此 Java 语言编程更符合人们的思维习惯。

3. Java 语言是分布式的

Java 从诞生之日就与网络联系在一起,Java 的主要优点是面向网络的编程,Java 支持网络应用程序的编程,使其成为一种分布式程序设计语言。Java 语言支持包括 HTTP 和 FTP 等基于 TCP/IP 的子库,在基本的 Java 网络编程中有一个网络应用编程接口,它提供了用于网络应用编程的类库,包括 URL、URLConnection、Socket、ServerSocket 等,使程序人员方便快捷地编写分布式应用程序。

4. Java 语言是健壮的

健壮性又称为稳定性。Java 语言在编译和执行过程中进行严格的语法检查,以减少错误的发生。Java 语言利用自动“垃圾”回收机制管理内存,防止程序员在管理内存时产生错误。使用 Java 异常处理机制捕获并响应异常情况,从而使程序在发生异常时能够继续运行。另外,Java 语言设计者在设计 Java 语言的过程中就已考虑如何减少编程过程中可能产生的错误。

5. Java 语言是安全的

Java 通常被用在网络环境中。为此,Java 提供了一个安全机制以防止恶意代码的攻击。除了 Java 语言具有的许多安全特性以外,Java 对通过网络下载类具有一个安全防范机制并提供安全管理机制。在执行 Java 程序过程中,Java 虚拟机对程序的安全性进行检测。一般来说,Java 程序是安全的,它不会访问或修改不允许访问的内存或文件。

6. Java 语言是体系结构独立的

Java 源代码不会针对一个特定平台进行编译,而是被转换成一种中间格式——字节码。字节码与体系结构无关,可以在任何有 Java 虚拟机的计算机上运行,而 Java 虚拟机与平台相关。Java 程序在虚拟机上运行;Java 虚拟机在操作系统上运行,用来解释和执行 Java 字节码。

只要安装 Java 虚拟机,就可以运行 Java 字节码,Java 虚拟机可以在各种平台上运行。

7. Java 语言是可移植的

这种可移植性来源于体系结构独立性,另外,Java 语言的设计目标就是让程序不用修改就可以在任何平台上运行。

8. Java 语言是解释型的

Java 程序在 Java 平台上被编译为字节码格式,然后可以在安装 Java 平台的任何系统中运行。在运行时,Java 平台中的 Java 解释器对这些字节码进行解释执行,执行过程中需要的类在连接阶段被载入到运行环境中。

这是 Java 语言的一个缺点,因为解释执行的语言一般会比编译执行的语言(如 C 和 C++)执行效率低。