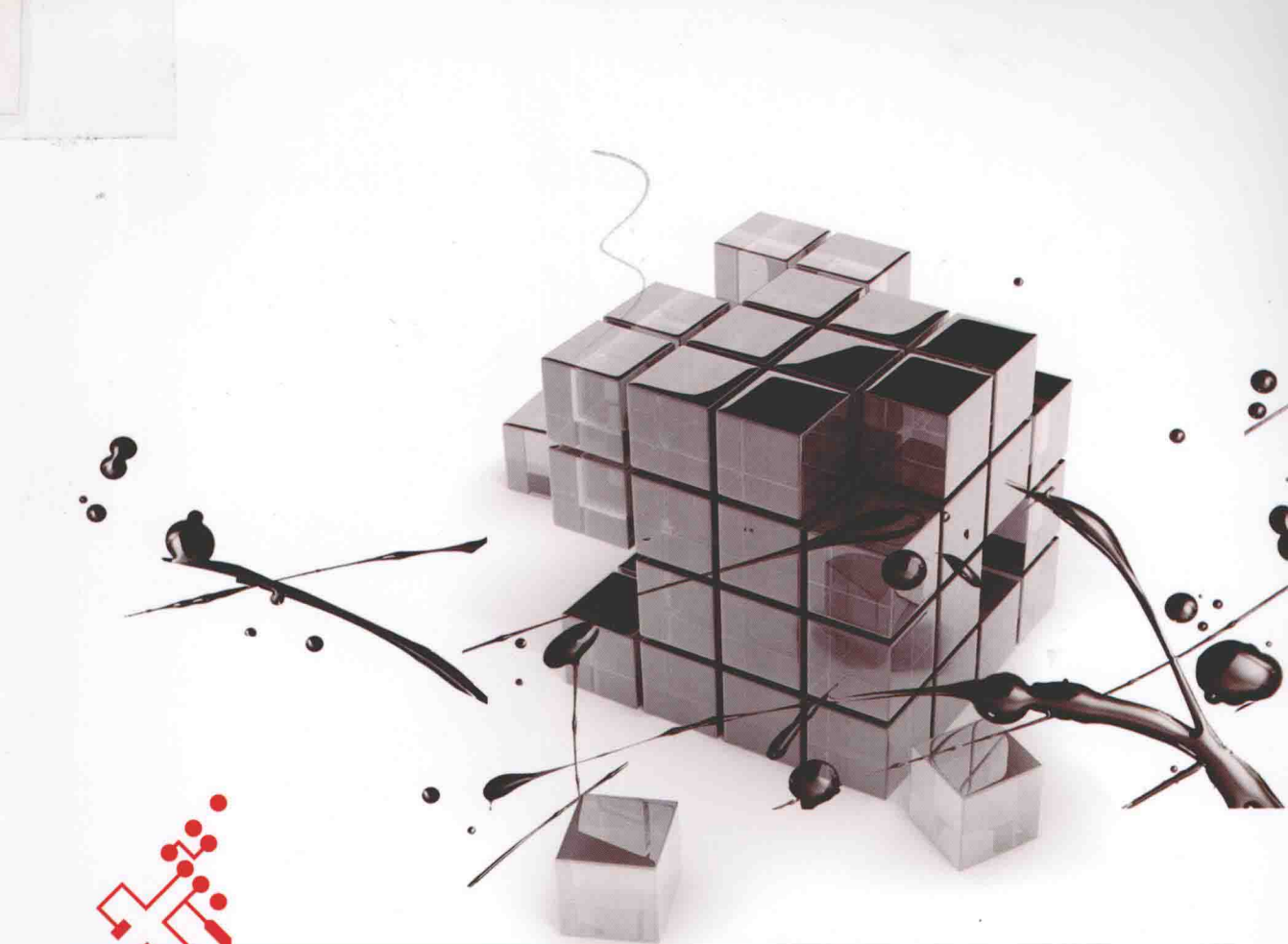




普通高等院校计算机基础教育“十三五”规划教材

Java

◆ 刘卫国 主编

语言程序设计

Java Yuyan Chengxu Sheji

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校计算机基

教材

Java 语言程序设计

刘卫国 主 编

李 激 副主编

ISBN 978-7-113-23247-8

ISBN 978-7-113-23247-8

I. ①J... II. ①刘... III. ①Java 语言—程序设计—教材 IV. ①TP317.8

中国版本图书馆CIP数据核字(2016)第197416号

中国版本图书馆CIP数据核字(2016)第197416号

作者: 刘卫国 主编
副主编: 李 激

责任编辑: 刘 颖 (010) 83208398

封面设计: 李 激
版式设计: 李 激
插图设计: 李 激
文字校对: 李 激
文字校对: 李 激

中国铁道出版社 (010) 83208398

http://www.criph.com

2016年8月第1版 2016年8月第1次印刷

本书定价: 49.00元

(010) 83208398

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

本书以 Java 语言作为实现工具,介绍程序设计的基本思想和方法。全书以实际问题的求解过程为向导,突出问题求解方法与思维能力训练。全书分为学习篇和实验篇。学习篇介绍 Java 语言的初步知识、面向对象程序设计的基本方法以及 Java 的实际应用,其内容包括概述,程序的数据描述,程序控制结构,类与对象,封装、继承与多态,异常处理与输入/输出,多线程,网络程序设计,数据库应用开发,Java 图形用户界面设计。实验篇提供实验指导,其中包括 14 个实验,每个实验都和课程学习内容相配合,以方便读者上机练习。

本书适合作为高等学校计算机程序设计课程的教材,也可供计算机应用人员阅读参考。

图书在版编目(CIP)数据

Java 语言程序设计 / 刘卫国主编. — 北京: 中国铁道出版社, 2016. 9

普通高等院校计算机基础教育“十三五”规划教材

ISBN 978-7-113-22247-5

I. ①J… II. ①刘… 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(2016)第 197416 号

书 名: Java 语言程序设计
作 者: 刘卫国 主编

策 划: 周海燕
责任编辑: 周海燕 彭立辉
封面设计: 乔 楚
责任校对: 汤淑梅
责任印制: 郭向伟

读者热线: (010) 63550836

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 北京海淀五色花印刷厂

版 次: 2016 年 9 月第 1 版 2016 年 9 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 22.25 字数: 497 千

书 号: ISBN 978-7-113-22247-5

定 价: 48.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 51873659



前言



计算机程序设计基础是高校计算机基础教学的核心课程,它以高级程序设计语言作为实现工具,介绍程序设计的基本思想和方法。通过本课程的学习,使学生掌握利用计算机解决问题的基本思路和方法,了解程序设计语言的组成要素和基本特征,具有编写程序和调试程序的基本技能。学习程序设计既可为学习后续相关计算机课程奠定基础,也有助于学生更加深刻地理解计算机求解问题的基本思想和方法,因而计算机程序设计课程也是计算思维能力培养的重要载体,应成为对学生进行计算思维能力培养最直接、最具操作性的课程。

Java 语言是目前流行的程序设计语言之一。随着 Internet 技术的发展,特别是 Web 应用开发技术的发展,Java 语言受到业界的广泛青睐,许多企业选择 Java 及其相关技术来构建它们的应用系统。这一方面是因为 Java 语言具有面向对象、平台无关性、可移植性、健壮性和安全性等优良特性;另一方面也因为 Java 技术对各个层次的分布式应用(包括嵌入式应用、桌面应用与企业级应用)都提供了全面系统的方法与技术。因此,目前 Java 已不仅仅是一种单纯的程序设计语言,而变成了一种支撑 Internet 应用的庞大技术体系,是一种开发软件的标准、规范与架构。事实上,学习 Java 本身也不仅仅在于学习如何使用 Java 语言的语法,更多的是学习如何应用 Java 所提供的资源与各种标准,以开发出架构更好、维护更容易的软件。

本书介绍 Java 语言的基础知识及程序设计的基本方法,全书以实际问题的求解过程为向导,突出从问题到算法,再到程序的一种思维过程,强调计算机求解问题的思路引导与程序设计思维方式的训练。全书分为学习篇和实验篇。学习篇是课程学习的主要内容,共分 10 章,其内容体系大致可分为三部分:第一部分为第 1 章~第 3 章,这一部分从对 Java 语言的初步认识入手,介绍 Java 语言的基本数据类型与运算、程序的 3 种基本结构及其实现方法,包括概述、程序的数据描述及程序控制结构;第二部分为第 4 章~第 6 章,这一部分是 Java 的核心与特色内容,介绍面向对象程序设计的概念和基本方法,包括类与对象,封装、继承与多态以及异常处理与输入/输出;第三部分为第 7 章~第 10 章,这一部分是对基础知识的扩充和发展,介绍 Java 的实际应用,包括多线程、网络程序设计、数据库应用以及 Java 图形用户界面设计。

学习 Java 语言程序设计,上机实践十分重要。只有通过上机实践,才能熟练掌握 Java 的语法知识,充分理解程序设计的基本思想和方法,并将所学知识应用到实际中。为了方便读者上机练习,实验篇中提供了 14 个实验,每个实验都和课程学习内容相配合,读者可以根据实际情况从每个实验中选择部分内容作为上机练习。实验内容既包含与 Java 语法规则相关的内容,也包含许多实际问题的程序设计,从而增强学生的学习兴趣,提高学生分析问题和解决问题的能力。

本书注重 Java 基础知识的介绍,结合 Java 流程控制语句介绍了程序设计的基本

知识和一些常用算法，因此本书可以作为第一门程序设计课程的教材使用。同时，本书也强调面向对象程序设计的核心技术和基本应用技术，因此也可供具有一定程序设计基础和经验的读者阅读参考。

本书由刘卫国任主编，李激任副主编。具体编写分工：第1章、第6~10章由刘卫国编写，第2、3章由童键编写，第4、5章由李激编写，实验篇由舒卫真编写。参与编写与讨论的还有彭胜、黎明、姚昱旻、柳小文、郭美、肖敏、李首洪、李博、邹奇林、周克涛、孙士闯、张娟、毛颖等。邹北骥教授、施荣华教授、彭小宁教授对本书的编写给予了许多指导，在此表示诚挚的感谢。

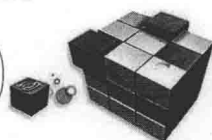
由于编者学识水平有限，书中的疏漏或不妥之处在所难免，恳请广大读者批评指正。

编者

2016年5月



目 录



学 习 篇

第 1 章 概述	2
1.1 Java 语言的发展历史与应用	2
1.1.1 Java 语言的发展历史	2
1.1.2 Java 语言的应用	4
1.2 Java 语言的特点	5
1.3 Java 程序的运行机制	8
1.3.1 Java 程序的分类	8
1.3.2 Java 程序的运行过程	8
1.3.3 Java 虚拟机	9
1.4 Java 程序的运行环境	10
1.4.1 Java 平台的版本选择	10
1.4.2 Java 开发工具——JDK	11
1.4.3 Java 集成开发环境——NetBeans	14
1.4.4 简单的 Java 程序	14
习题	20
第 2 章 程序的数据描述	22
2.1 Java 符号集	22
2.1.1 Java 中的标识符	22
2.1.2 Java 的關鍵字	23
2.1.3 Java 的分隔符	23
2.2 常量和变量	24
2.2.1 常量	25
2.2.2 变量	25
2.3 基本数据类型	26
2.3.1 整型	27
2.3.2 浮点型	28
2.3.3 字符型	28
2.3.4 布尔型	29
2.3.5 基本数据类型值间的转换	29
2.3.6 基本数据类型	30
2.4 常用数值计算方法	32



2.5	运算符和表达式	33
2.5.1	赋值运算	33
2.5.2	算术运算	34
2.5.3	关系运算与逻辑运算	35
2.5.4	位运算符	38
2.5.5	其他运算	40
2.5.6	运算符的优先级及结合性	41
2.6	字符串	41
2.6.1	String 类	41
2.6.2	StringBuffer 类	44
2.7	数组	46
2.7.1	数组的概念与创建	46
2.7.2	数组的引用	47
	习题	48
第 3 章	程序控制结构	52
3.1	程序设计概述	52
3.1.1	程序设计的基本步骤	52
3.1.2	算法及其描述	53
3.2	顺序结构程序	58
3.2.1	Java 语句	58
3.2.2	基本输入/输出	59
3.2.3	顺序结构程序举例	61
3.3	选择结构程序	62
3.3.1	单分支条件语句	62
3.3.2	双分支条件语句	63
3.3.3	多分支条件语句	64
3.3.4	switch 语句	66
3.3.5	选择结构程序举例	67
3.4	循环结构程序	69
3.4.1	for 循环语句	69
3.4.2	while 循环语句	71
3.4.3	do...while 循环语句	72
3.4.4	嵌套循环	73
3.4.5	循环结构程序举例	75
3.5	跳转语句	77
3.5.1	break 语句	77
3.5.2	continue 语句	78
	习题	79
第 4 章	类与对象	83
4.1	面向对象的基本概念	83



4.1.1	对象、类和消息	83
4.1.2	面向对象的特性	84
4.2	Java 类	85
4.2.1	类的基本结构	85
4.2.2	类的成员	86
4.2.3	构造方法	88
4.3	Java 对象	89
4.3.1	对象的创建和初始化	89
4.3.2	对象的引用和清除	92
4.3.3	方法的调用	95
4.4	包	99
4.4.1	包的概念	99
4.4.2	包的创建与使用	100
	习题	101
第 5 章	封装、继承与多态	104
5.1	封装	104
5.1.1	封装的概念	104
5.1.2	访问控制符	105
5.2	继承	108
5.2.1	继承的概念	108
5.2.2	继承的应用	108
5.2.3	this 关键字与 super 关键字	113
5.2.4	继承的终止	118
5.3	多态	118
5.3.1	方法重载	119
5.3.2	方法覆盖与变量多态	120
5.3.3	抽象类	121
5.4	接口	125
5.4.1	接口的概念	125
5.4.2	接口的应用	126
5.4.3	抽象类与接口的比较	127
	习题	128
第 6 章	异常处理与输入/输出	131
6.1	异常处理	131
6.1.1	异常的概念	132
6.1.2	异常的处理	132
6.1.3	异常类的层次结构	134
6.1.4	自定义异常	135
6.1.5	异常处理的使用原则	137



6.2	Java 的 I/O 处理	137
6.2.1	I/O 流的概念	138
6.2.2	基本的 I/O 类	138
6.2.3	实体流与过滤流	141
6.2.4	标准输入/输出对象	142
6.3	字节流应用	142
6.3.1	InputStream 类和 OutputStream 类	143
6.3.2	常用实体流	145
6.3.3	常用过滤流	147
6.3.4	字节流应用举例	147
6.4	字符流应用	149
6.4.1	Reader 类和 Writer 类	150
6.4.2	字符流中的实体流和过滤流	151
6.4.3	InputStreamReader 类和 OutputStreamWriter 类	151
6.4.4	字符流应用举例	152
6.5	文件处理	153
6.5.1	File 类	153
6.5.2	文件的顺序读/写	155
6.5.3	文件的随机访问	159
	习题	162
第 7 章	多线程	166
7.1	进程与线程	166
7.1.1	进程的概念	166
7.1.2	线程的概念	167
7.2	线程的创建	168
7.2.1	继承 Thread 类	169
7.2.2	实现 Runnable 接口	169
7.2.3	两种创建方式的比较	170
7.2.4	线程的生命周期	171
7.3	线程的调度与同步机制	171
7.3.1	线程的调度	172
7.3.2	线程的同步	177
7.4	线程的通信	180
7.5	多线程应用举例	183
	习题	186
第 8 章	网络程序设计	190
8.1	网络程序设计基础知识	190
8.1.1	TCP/IP 简介	190
8.1.2	地址、域名与端口号	191



8.1.3	InetAddress 类	192
8.2	URL 编程	192
8.2.1	URL 类	192
8.2.2	URLConnection 类	195
8.3	Socket 编程方法	197
8.3.1	Socket 的概念	197
8.3.2	Socket 编程方法	198
8.4	Datagram 编程	203
8.4.1	Datagram 的概念	203
8.4.2	Datagram 编程的方法	204
	习题	211
第 9 章 数据库应用开发		215
9.1	JDBC 概述	215
9.1.1	数据库的基本概念	215
9.1.2	JDBC 的原理	217
9.2	数据库访问	218
9.2.1	JDBC 的常用接口和类	219
9.2.2	JDBC 数据库连接	220
9.2.3	JDBC 数据库操作	224
	习题	236
第 10 章 Java 图形用户界面设计		239
10.1	图形用户界面设计概述	239
10.1.1	Java 图形用户界面工具	239
10.1.2	Java 图形用户界面的构成	241
10.2	容器和布局管理器	242
10.2.1	Swing 的常用容器	242
10.2.2	常用的布局管理器	250
10.3	Swing 的常用控件	254
10.3.1	Swing 标签和文本框	254
10.3.2	Swing 命令按钮	257
10.3.3	Swing 的组合框和列表框	258
10.3.4	Swing 复选框和单选按钮	259
10.3.5	Swing 中的高级控件	262
10.4	使用 Netbeans 开发简单的 GUI 应用	269
10.4.1	Netbeans 的 GUI 生成器	269
10.4.2	Netbeans 的 GUI 开发实例	270
	习题	274



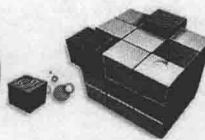
实 验 篇

实验 1 简单的 Java 程序.....	278
实验 2 程序的数据描述.....	281
实验 3 顺序结构程序.....	284
实验 4 选择结构程序.....	287
实验 5 循环结构程序.....	291
实验 6 字符串处理与数组应用.....	295
实验 7 Java 类与对象.....	299
实验 8 类的继承和多态机制.....	302
实验 9 异常处理.....	308
实验 10 输入/输出处理.....	314
实验 11 多线程编程.....	322
实验 12 网络程序设计.....	327
实验 13 数据库应用开发.....	333
实验 14 图形用户界面设计.....	338
参考文献.....	344

第 10 章 Java 图形用户界面设计.....	339
10.1 图形用户界面设计概述.....	339
10.1.1 Java 图形用户界面设计工具.....	339
10.1.2 Java 图形用户界面的特点.....	341
10.2 容器和窗口管理.....	342
10.2.1 Swing 的常用容器.....	342
10.2.2 常用的窗口管理器.....	343
10.3 Swing 的常用控件.....	344
10.3.1 Swing 的文本域.....	344
10.3.2 Swing 的按钮.....	345
10.3.3 Swing 的列表框和下拉菜单.....	346
10.3.4 Swing 的复选框和单选按钮.....	347
10.3.5 Swing 中的高级控件.....	348
10.4 使用 NetBeans 开发简单的 GUI 应用.....	349
10.4.1 NetBeans 的 GUI 生成器.....	349
10.4.2 NetBeans 的 GUI 开发案例.....	350

学习篇

学习篇是课程学习的主要内容，共分 10 章，其内容体系大致分为三部分：第一部分为第 1 章~第 3 章，这一部分从对 Java 语言的初步认识入手，介绍 Java 语言的基本数据类型与运算、程序的 3 种基本结构及其实现方法，包括概述、程序的数据描述及程序控制结构；第二部分为第 4 章~第 6 章，这一部分是 Java 的核心与特色内容，介绍面向对象程序设计的概念和基本方法，包括类与对象，封装、继承与多态以及异常处理与输入/输出；第三部分为第 7 章~第 10 章，这一部分是对基础知识的扩充和发展，介绍 Java 的实际应用，包括多线程、网络程序设计、数据库应用开发以及图形用户界面开发技术。



在经历了以大型机为代表的集中计算模式和以 PC 为代表的分散计算模式之后, Internet 的出现使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机是异构的, 即计算机的类型和操作系统是不一样的。网络计算模式的另一个特点是代码可以通过网络在各种计算机上进行迁移, 这就迫切需要一种跨平台的程序设计语言, 使得用它编写的程序能够在网络中的各种计算机上正常运行。Java 就是在这种需求下应运而生的。

随着 Internet 技术的发展, 特别是 Web 应用开发技术的发展, Java 语言受到业界的广泛青睐, 许多企业选择 Java 及其相关技术来构建它们的应用系统。对于进行 Internet 相关的程序开发, Java 语言是最佳选择。

学习目标:

- 了解 Java 语言的发展历史和现状。
- 理解 Java 语言的自身特点和 Java 程序的运行机制。
- 掌握 Java 开发工具 (JDK) 的安装与配置。
- 能使用文本编辑器和 Java 集成开发环境 (IDE) 编写和运行简单的 Java 程序。



1.1 Java 语言的发展历史与应用

Java 语言是美国 Sun 公司推出的新一代面向对象程序设计语言, 而 Java 本是印度尼西亚的一个小岛——爪哇岛, 盛产咖啡, 将两者联系在一起有着很有趣的故事。当初 Java 语言还不为世人所熟知, 几位项目组的成员在讨论给这个新的语言取什么名字, 当时他们正在喝一种来自爪哇岛的咖啡, 有人灵机一动说就叫 Java 怎样, 很快得到了其他人的赞赏, 于是 Java 这个名字就这样传开了, 寓意为世人端上一杯热咖啡。现在看来, Java 这杯咖啡已经飘香世界各地。

1.1.1 Java 语言的发展历史

Java 语言来源于 Sun 公司的一个叫作 Green 的项目。早在 1991 年, Sun 公司启动了 Green 项目, 该项目最初的目的是开发一种分布式系统架构, 使其能够在各种智能家电上运行。由于该项目组的成员都具有深厚的 C++ 开发背景, 所以一开始准备使用 C++ 语言来完成这个项目, 但很快便意识到 C++ 过于复杂和庞大且安全性差, 并不适用于这个项目, 有必要为这个项目专门研发一种新的程序设计语言, 于是对一门新语



言的研究就这样展开了。最后项目组成员基于 C++ 开发了一种新的语言 Oak (橡树), 这就是 Java 语言的前身。

Oak 语言是一种用于网络的精巧而安全的语言。该语言采用了许多 C++ 语言的语法, 提高了安全性, 并且是面向对象的语言, 但是 Oak 语言在商业上并未获得成功。直至 1994 年, 随着 Internet 的迅猛发展和 WWW 应用的快速增长, Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点非常符合 Internet 的需要, 于是改进了该语言的设计, 实现了如下几个目标:

(1) 创建一种面向对象的程序设计语言, 而不是面向过程的语言。

(2) 提供一个解释执行的程序运行环境, 使程序代码独立于计算机系统平台。

(3) 吸收 C 和 C++ 的优点, 使程序员容易掌握。

(4) 去掉 C 和 C++ 中影响程序健壮性的部分, 使程序更安全, 例如指针、内存申请和释放等。

(5) 实现多线程, 使得程序能够同时执行多个任务。

(6) 提供动态下载程序代码的机制。

(7) 提供代码校验机制以保证安全性。

最终, 更为完善的 Oak 版本被重新起了一个更响亮的名字——Java。1995 年春, Sun 公司发表了 Java 的完整技术规范, 立即得到包括 Netscape 公司在内的各 WWW 厂商的广泛支持。不久, 许多著名的大公司, 如 IBM、Novell、Oracle、SGI 和 Borland 公司等都相继购买了 Java 的使用权, Java 得到了广泛的支持。Bill Gates 在反复仔细研究了 Java 的技术规范之后, 认为“Java 确实是最卓越的程序设计语言”, 继而 Microsoft 公司购买了 Java 的使用权并很快推出了集成化的 Java 开发环境——Visual J++, 至此 Java 在软件界的地位不可动摇。

Java 语言的诞生在日新月异的 IT (Information Technology, 信息技术) 领域算是一个新生事物, Java 本身一直没有停下变革、创新脚步, Java 现在已经不单单是一种计算机语言, 很多情况下它是一个平台、一个系统。

1995 年 Java 公开亮相后, 经过一年的试用和改进, Java 1.0 版在 1996 年年初正式发表。这个版本包括两部分: Java 运行环境 (Java Runtime Environment, JRE) 和 Java 开发工具 (Java Development Kit, JDK)。在 Java 运行环境中包括了核心 API (Application Program Interface)、集成 API、用户界面 API、发布技术、Java 虚拟机 (Java Virtual Machine, JVM) 5 个部分, 而 Java 开发工具还包括了编译 Java 程序的编译器 (即 javac)。在 JDK 1.0 时代, JDK 除了 AWT (一种用于开发图形用户界面的 API) 外, 其他的库并不完整。

1997 年 2 月发布了 JDK 1.1, 1998 年 12 月发布了 JDK 1.2 版本 (Sun 公司觉得 Java 改变足够大而将 Java 语言版本号提升为 2.0, 简称 Java 2), 1999 年 6 月 Sun 公司发布 Java 的 3 个版本: 标准版 (Java 2 Standard Edition, J2SE)、企业版 (Java 2 Enterprise Edition, J2EE) 和微型版 (Java 2 Micro Edition, J2ME), 表明了 Java 开始向桌面应用、企业应用和移动设备应用三大领域渗透。

2000 年 5 月, JDK 1.3 和 JDK 1.4 先后发布。2001 年 9 月, J2EE 1.3 发布。2002 年 2 月, J2SE 1.4 发布, 自此 Java 的计算能力有了大幅提升。2004 年 9 月, J2SE 1.5



发布, 该版本扩充了大量的 API 类库并且引进了诸多新的特性, 它的发布是 Java 语言发展史上的又一里程碑事件。为了表示这个版本的重要性, J2SE 1.5 更名为 J2SE 5.0。

2005 年 6 月, Sun 公司发布 Java SE 6, 该版本主要的改进是性能的提升, 并且强化了对各种脚本语言的支持。此时, Java 的各种版本已经更名以取消其中的数字“2”, J2SE 更名为 Java SE, J2EE 更名为 Java EE, J2ME 更名为 Java ME。

2006 年 11 月, Sun 公司宣布 Java 全线采纳 GNU General Public License Version 2, 从而开始有步骤地公开 Java 的源代码。2007 年 12 月, Sun 公司正式发布 Java 开源集成开发环境 (Integrated Development Environment, IDE)——NetBeans 6.0。

2008 年 9 月, 第一届 JVM 语言峰会的召开标志着 Java 从一个以语言为中心的技术转变成为一个以平台为中心的技术, 基于 JVM 的动态语言达 240 多种, 这其中以 Jython、Ruby 的发展最为迅猛。

2009 年 4 月, Oracle 公司以 74 亿美元收购了 Java 的缔造者——Sun 公司。Oracle 公司负责产品开发的执行副总裁 Thomas Kurian 称 Java 为王冠上的一颗明珠, 他示意将会增强和扩展 Java 编程模型, 支撑各种新兴的应用开发模式。

2011 年 7 月, Oracle 公司发布 Java SE 7, 该版本主要改进了模块化的功能及 JVM 对动态语言的支持, 新增了功能特性, 性能又一次得到提升。

2014 年 3 月, Oracle 发布 Java SE 8, 该版本主要新增了 Lambda 表达式和 Nashorn 引擎。新增 Lambda 表达式使 Java 更易于为多核处理器编写代码, 新增 Nashorn 使 Java 程序可以和 JavaScript 代码互操作。纵观 Java 的历史, 人们有理由相信, 由于强强结合, Java 这棵在 IT 业 20 年屹立不倒的苍天大树在易主之后会焕发新的生机, 迎来蓬勃发展的又一春天。

1.1.2 Java 语言的应用

任何一种程序设计语言的诞生都有两个不可缺少的条件: 一是行业的需要; 二是语言本身的优势。Java 是一种基于 Internet 的网络程序设计语言, 具有十分广泛的市场前景, 它改变了传统的计算机应用方式, 在安全性能和与硬件平台无关性等方面具有极大的优势。面向对象的编程环境、对多媒体信息的全面支持、多线程技术的使用都促进了 Java 语言的普及。目前它在各个重要的行业部门得到了广泛的应用, 而且出现在各种各样的设备、计算机和网络中。从笔记本式计算机到数据中心, 从游戏控制台到科学计算超级计算机, 从手机到 Internet, Java 无处不在。

从权威的 Tiobe 全球程序设计语言使用和关注度排行来看, 在最近 10 年的绝大部分时间里, Java 以较大优势位居所有程序设计语言的首位。2016 年 5 月份排行榜中, Java、C、C++ 依然是前三甲 (见 <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>)。

Java 语言有着广泛的应用前景, 大体上可以分成以下几方面:

- (1) 所有面向对象的应用开发, 包括面向对象的事件描述、处理和综合等。
- (2) 计算过程可视化以及可操作化软件的开发。
- (3) 动态画面的设计, 包括图形图像的调用。
- (4) 交互操作的设计, 包括选择交互、定向交互、控制流程等。



(5) Internet 的系统管理功能模块的设计,包括 Web 页面的动态设计、管理和交互操作设计等。

(6) Intranet (企业内部网)上的软件开发(直接面向企业内部用户的软件)。

(7) 与各类数据库连接查询的 SQL 语句实现。

(8) 网络通信与移动通信以及网络集成方面。



1.2 Java 语言的特点

Java 语言自产生以来, Sun 公司致力于将 Java 设计成为一种简单的、面向对象的、分布式的、解释执行的、健壮的、安全的、平台无关的、可移植的、高性能的、多线程的和动态的语言,正是这些特点,使得 Java 成为目前使用最为广泛的网络程序设计语言。

1. 简单

这里的简单当然不是功能上的简单,而是指 Java 语言在学习、使用和维护上的方便。Java 的简单性是与 C/C++ 比较得出的结论, Java 是在 C/C++ 的基础上开发的,它继承了 C/C++ 的许多特性,但同时也摒弃了 C/C++ 中烦琐的、难以理解的或存在安全隐患的功能,比如指针、运算符重载和多重继承等,并对经常使用的操作进行了简化,比如字符串的操作和网络编程等。由于 Java 在设计时参考了许多 C/C++ 的语法和特性,使得无论是对程序语言的初学者还是对有一定基础的 C/C++ 程序员来说, Java 都是易学、易用和易维护的。

2. 面向对象

现实世界是由各种各样的实体(事物、对象)所组成的,每种对象都有自己的内部状态和运动规律,不同对象间的相互联系和相互作用就构成了各种不同的系统,进而构成整个客观世界。人们为了更好地认识客观世界,把具有相似内部状态和运动规律的实体(事物、对象)综合在一起称为类。类是具有相似内部状态和运动规律的实体的抽象,进而人们抽象地认为客观世界是由不同类的事物间相互联系和相互作用所构成的一个整体。计算机软件的目的就是模拟现实世界,使现实世界中各种不同的系统在计算机中得以实现,进而为人们的工作、学习、生活提供帮助。

面向对象(Object-Oriented, OO)其实是现实世界模型的自然延伸,它是按人们认识客观世界的系统思维方式,采用基于对象(实体)的概念建立模型、模拟客观世界分析、设计和实现软件的办法。通过面向对象的观念使计算机软件系统能与现实世界中的系统一一对应。现实世界中的对象均有属性和行为,反映到计算机程序上,属性则表示对象的数据,行为表示对象的方法(其作用是处理数据或同外界交互)。如果说传统的面向过程程序设计语言是以过程为中心、以算法为驱动,面向对象程序设计语言则是以对象为中心、以消息为驱动。用公式表示,面向过程程序设计语言为:程序=算法+数据;面向对象程序设计语言为:程序=对象+消息。

Java 语言是面向对象的程序设计语言,其中的绝大部分语法元素都是对象。另外, Java 与 C++ 相比取消了复杂的多重继承,取而代之的接口概念使得 Java 更加容易使用。

3. 分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机



上,操作分布是指把一个计算分散在不同主机上处理。Java 支持 WWW 客户机/服务器 (Client/Server) 计算模式,因此,它支持这两种分布性。对于前者,Java 提供了一个叫作 URL 的对象,利用这个对象,可以打开并访问具有相同 URL 地址的对象,访问方式与访问本地文件系统相同。对于后者,Java 的 Applet 小程序可以从服务器下载到客户端,即部分计算在客户端进行,提高系统执行效率。Java 提供了一整套网络类库,开发人员可以利用类库进行网络程序设计,方便地实现 Java 的分布式特性。

4. 解释执行

Java 是解释执行的语言。高级语言程序必须转换为机器语言程序才能执行,但不同的计算机系统所使用的机器语言不同。Java 为了实现“编写一次,随处运行”(Write Once,Run Anywhere)的目标,Java 的源程序在编译时,并不直接编译成特定的机器语言程序,而是编译成与机器结构无关的字节码(Byte Code),然后,这种字节码在任何 Java 的运行环境中由 Java 虚拟机(JVM)解释执行。

JVM 能直接在任何机器上执行,为字节码提供运行环境。当 JVM 解释执行 Java 程序时,Java 实时编译器(Just In Time, JIT)会将字节码译成目标平台对应的机器语言的指令代码。

5. 健壮性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。它提供自动垃圾回收机制(Garbage Collection)来进行内存管理,防止程序员在管理内存时容易产生的错误。通过集成的面向对象的异常(Exception)处理机制,在编译时,Java 提示出可能出现但未被处理的异常,帮助程序员正确地进行选择以防止系统的崩溃。另外,在编译时 Java 还可捕获类型声明中许多常见的错误,防止动态运行时不匹配问题的出现,保证程序更加可靠。

6. 安全性

由于 Java 主要用于网络应用程序开发,因此对安全性有较高的要求。如果没有安全保证,用户从网络下载程序执行就非常危险。Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。Java 的安全性可从两方面得到保证:一方面,Java 不支持指针和内存释放等 C++所具有的功能,一切对内存的访问都必须通过对象的实例变量来实现,这样就避免了非法内存操作和指针操作时容易产生的错误;另一方面,类装载通过将本机类和网络资源类的名称分开,这样,调入类时总要经过检查,可以防止用欺骗手段访问对象的私有成员,从而保证了安全性。

7. 平台无关性

所有软件都有其运行的特定环境,就是所谓的平台,平台无关有两种:源代码级和目标代码级。C/C++具有一定程度的源代码级平台无关,表明用 C/C++写的应用程序不用修改只需重新编译就可以在不同平台上运行,而 Java 的平台无关性是指用 Java 写的应用程序不用修改就可在不同的软硬件平台上运行,这主要靠 Java 虚拟机(JVM)在目标代码级实现。JVM 是一种抽象机器,它附着在具体操作系统之上,本身具有一套虚拟机指令,并有自己的栈、寄存器组等。JVM 是 Java 平台无关的基础,也是 Java 程序可移植性的保证,在 JVM 上,有一个 Java 解释器用来解释 Java 编译器编译后的