

21世纪高等院校计算机教材系列

实用Java

程序设计教程

● 宛延闯 鲁玛勒 定海 等编著

 机械工业出版社
CHINA MACHINE PRESS



实用 Java 程序设计教程

宛延阁 鲁玛勒 定海 等编著

机械工业出版社

本教程以面向对象基本概念为主干线,用最新的 JDK 开发软件介绍了从简单的 Java 程序,到用 Java 语言编写出漂亮的动画小应用、网页和具有信息综合处理能力的实用 Java 应用程序。

本书深入浅出、通俗流畅,所有的 Java 实用概念均通过大量典型范例循序渐进地引入,书中还配有生动的插图。全书共分 15 章 4 个附录,每章均有习题,读者可参考使用。

本书可作为高等院校学生学习 Java 程序设计的教材,也是广大 Java 爱好者和科技工作者学习 Java 程序设计必备的参考书。

图书在版编目(CIP)数据

实用 Java 程序设计教程/宛延圉等编著. —北京:机械工业出版社, 2004.6

(21 世纪高等院校计算机教材系列)

ISBN 7-111-13927-5

I. 实... II. 宛... III. Java 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 006700 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:孙 业

责任印制:李 妍

北京机工印刷厂印刷·新华书店北京发行所发行

2004 年 6 月第 1 版·第 1 次印刷

787mm×1092mm¹/₁₆·23.25 印张·574 千字

0 001—5 000 册

定价:32.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68993821、88379646

封面无防伪标均为盗版

出版说明

计算机技术是一门迅速发展的现代科学技术,它在经济建设与社会发展中,发挥着非常重要的作用。近年来,我国高等院校十分注重人才的培养,大力提倡素质教育、优化知识结构,提倡大学生必须掌握计算机应用技术。为了满足教育的需求,机械工业出版社组织了这套“21世纪高等院校计算机教材系列”。

在本套系列教材的组织编写过程中,我社聘请了各高等院校相关课程的主讲老师进行了充分的调研和细致的研讨,并针对非计算机专业的课程特点,根据自身的教学经验,总结出知识点、重点和难点,一并纳入到教材中。

本套系列教材定位准确,注重理论教学和实践教学相结合,逻辑性强,层次分明,叙述准确而精炼,图文并茂,习题丰富,非常适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班和自学用书。

参加编写本系列教材的院校包括:清华大学、西安交通大学、北方交通大学、北京邮电大学、北京化工大学、北京科技大学、山东大学、首都经贸大学等。

机械工业出版社

前 言

人类已经步入 21 世纪,21 世纪将是科学技术高速发展的信息时代,计算机是处理信息的主要工具。人们已经认识到,计算机知识已经成为人类当代文化的一个重要组成部分。

计算机诞生 50 多年来,其应用方式经历了主机时代、微型计算机时代、计算机网络时代三个发展阶段。网络将是计算机应用的极为重要的领域。因为单机的应用是有限的,而网络的应用是无限的。人们要使用计算机必然会同时用到网络。21 世纪不会使用网络就等于不会使用计算机。多媒体技术更是使计算机不仅能处理文字信息,而且能有效地处理文字、图形、图像、声音等各种信息,使计算机更加贴近人们的生活,能够更好地模拟人们活动的各种形式。

当前,Java 已成为把网络 and 多媒体技术融合到一起的网上语言。Java 诞生于 20 世纪 90 年代中叶。Java 在计算机软件领域引起的革命,使它一跃成为在万维网(WWW)上开发软件的主要工具,并正在迅速取代 Pascal,C 和 C++ 的地位。

根据当前因特网(Internet)飞速发展的要求,Java 已经不仅仅是网上语言,而且逐渐形成了一门在因特网上不可或缺的技术。Java 是有史以来惟一被工业界和学术界同时认可的语言。Java 从 C++ 而来,去粗取精,升华提高,脱胎换骨。学习程序设计语言与学习打球、练武术一样,根基要正,Java 应该是初学者的第一语言。

本书是作者 1997 年至 2003 年期间,在中国科学院研究生院、中国科学技术大学研究生院和中国科学院研究生院远程与继续教育学院开设的“Java 程序设计”课程讲稿的基础上,对其进行充实、加工、再整理而成,是理论、实践、教学三结合的产物,能够帮助读者较快地正确地掌握 Java 语言和技术。

本书是一本实用教科书,是以面向对象程序设计为主干线,深入浅出地介绍 Java 基本概念、基本方法及技巧、Java 虚拟机和 Java 的各种实用技术,应用最新的 Java 开发套件(Java Developer Kit—JDK),编制有动画效果的实用小应用和实用的 Java 应用程序,使之能够与因特网上各种信息领域的需求接轨。

读者只需具有最基本的计算机知识,便能通过本书循序渐进地学会 Java 程序设计,每章结尾都留有练习题,它们是学习编程的关键。如果不编程,就永远学不会 Java 语言。我们希望读者很快就会发现编制 Java 程序是一件非常有趣的事情。

计算机科学技术以惊人的速度向前发展,各行各业的人都迫切要求学习计算机知识,即使是已有一定基础的计算机应用人员,也需要不断地学习新知识,进行知识更新。为此,我们可以一点不夸张地说,掌握 Java 技术是衡量软件工程师是否紧跟国际前沿技术的一种标志,也是学生走向社会能力的标志。

全书共分 15 章 4 个附录。第 1 章 Java 和 Web;第 2 章面向对象程序设计和软件开发;第 3 章你的第一个 Java 程序;第 4 章 Java 语言基础;第 5 章类和对象;第 6 章数组和字符串;第 7 章子类、继承和多态;第 8 章例外处理;第 9 章创建图形用户界面;第 10 章多线程;第 11 章 Applet 技术基础;第 12 章音乐和动画;第 13 章 Java 高级编程;第 14 章杂类;第 15 章实例剖析

——运动会计分系统。每章均有习题。为了方便读者学习和教师的教学,与本书配套的《实用 Java 程序设计教程习题解答及上机实习》一书也随之出版。为了更快更好地使读者掌握 Java 语言和技术,我们还编著有《Java 编程技巧》(配光盘)一书不久也将由机械工业出版社出版。

参加本书编写的还有苏文、宛霞、李保林、崔柯、乔立琴、蔡凤奇、李石乔、米慧敏和石良秀。甄炜、王浩枫和韩文智参加了本书的整理和校对工作。在编写过程中得到了中国科学院研究生院的大力支持,在此一并表示感谢!

宛延阁

目 录

出版说明

前言

第 1 章 Java 和 Web	1
1.1 计算机和万维网	1
1.2 程序设计语言和 Java	2
1.3 Java 与 C,C++	3
习题	6
第 2 章 面向对象程序设计和软件开发	8
2.1 历史发展的必然	8
2.2 对象、类和消息	8
2.3 抽象	9
2.4 封装	10
2.5 继承	10
2.6 多态	11
2.7 软件工程和软件生命周期	12
2.8 面向对象程序设计	13
习题	15
第 3 章 你的第一个 Java 程序	16
3.1 JDK 的获得与安装	16
3.2 第一个应用程序	17
3.3 第一个小应用	18
3.3.1 小应用的改进(一)	21
3.3.2 小应用的改进(二)	22
3.3.3 Drawing 方法小结	22
3.4 小应用与应用程序	23
3.4.1 Java 应用程序的结构	24
3.4.2 小应用的结构	24
3.5 Java 结构和特点	25
3.5.1 Java 的特点	25
3.5.2 Java 虚拟机	26
3.5.3 Java 运行系统运作过程	27
3.6 Java 规范	28
3.7 Java 运行系统具有很好的稳定性	29
3.8 确保 Java 的高度安全性	29

3.9 Java 技术的未来	30
习题	31
第 4 章 Java 语言基础	32
4.1 标识符和关键字	32
4.2 基本数据类型	33
4.2.1 整数类型	34
4.2.2 浮点数类型	34
4.2.3 布尔类型	35
4.2.4 字符类型	35
4.3 常量	35
4.3.1 布尔常量	35
4.3.2 整型常量	35
4.3.3 字符常量	36
4.3.4 字符串常量	37
4.3.5 浮点常量	37
4.3.6 其他文字常量	37
4.4 变量	37
4.4.1 举例	38
4.4.2 值得注意的几个问题	39
4.5 表达式,语句和块	40
4.6 运算符	41
4.6.1 赋值运算符	41
4.6.2 赋值语句	41
4.6.3 算术运算符	43
4.6.4 关系运算符	45
4.6.5 逻辑运算符	46
4.6.6 位运算符	47
4.6.7 条件赋值运算符	47
4.6.8 广义赋值运算符	48
4.6.9 对象运算符	48
4.6.10 括号与方括号运算符	48
4.7 运算符的优先级	48
4.8 类型转换	49
4.8.1 自动类型转换	49
4.8.2 强制转换	50
4.9 注释	50
4.10 流控制语句	51
4.10.1 if-else 语句	51
4.10.2 switch 语句	52
4.10.3 循环语句	53
4.10.4 转移语句	56

4.10.5 递归	58
习题	59
第5章 类和对象	63
5.1 类的定义	63
5.2 对象	65
5.2.1 对象生成	66
5.2.2 对象使用	69
5.2.3 释放对象	70
5.3 方法	71
5.3.1 实现类行为的方法	71
5.3.2 人民币与美元相互兑换实例分析	72
5.3.3 构造方法	75
5.3.4 访问权限修饰符:public 和 private	77
5.3.5 局部变量、返回值和参数传递	80
5.4 static 变量与方法	84
习题	86
第6章 数组和字符串	89
6.1 数组	89
6.1.1 基本数据类型的数组	89
6.1.2 对象数组	93
6.2 把数组传送给方法	98
6.3 this 引用	100
6.3.1 this 引用的第一种场合	100
6.3.2 this 引用的第二种场合	101
6.3.3 this 引用的第三种场合	102
6.4 多维数组	103
6.4.1 声明和创建二维数组	103
6.4.2 二维数组的引用	104
6.4.3 二维数组的初始化	104
6.5 不变字符串:String 类	105
6.5.1 字符串的长度	106
6.5.2 查找字符	107
6.5.3 字符串比较	108
6.5.4 子串	109
6.5.5 修改字符串	109
6.5.6 toString()方法及其他有用的 String 方法	110
6.5.7 String 类的构造方法	110
6.6 基本数据类型与引用数据类型	111
6.7 可变字符串:StringBuffer 类	115
6.7.1 StringBuffer 类的构造方法	115
6.7.2 StringBuffer 类的其他有用方法	115

6.8	向量	117
6.8.1	向量的声明	118
6.8.2	Vector 类的方法	118
6.9	堆栈	120
6.10	Dictionary 抽象类和它的散列表实现	121
6.10.1	Dictionary 抽象类	122
6.10.2	散列表	122
6.11	有关数据结构的其他知识	123
	习题	126
第 7 章	子类、继承和多态	129
7.1	子类和继承	129
7.1.1	成员变量的隐藏和方法重写	129
7.1.2	super 引用	131
7.1.3	继承	132
7.1.4	方法重载与方法重写	133
7.2	运行时多态	134
7.3	重载构造方法	136
7.4	继承和构造方法	137
7.5	用 this 和 super 调用其他构造方法	140
7.6	抽象类和抽象方法	142
7.7	Java 的名字空间和包	143
7.7.1	Java 的名字空间	144
7.7.2	Java 提供的包	144
7.7.3	定义自己的包	145
7.7.4	使用 Java 类库的方式	145
7.7.5	包的引用	146
7.7.6	存取保护	146
7.8	对象类型转换	147
7.9	实例成员和类成员	148
7.9.1	实例变量和类变量在使用上的区别	148
7.9.2	实例方法和类方法在使用上的区别	148
7.9.3	实例成员和类成员使用小结	149
7.10	final 类和方法	149
7.11	域	150
7.11.1	静态域	150
7.11.2	静态初始化块	150
7.11.3	最终域	151
7.11.4	易失域	151
7.12	方法	151
7.12.1	抽象方法	152
7.12.2	静态方法	152

7.12.3 最终方法	152
7.12.4 本地方法	152
7.12.5 同步方法	153
7.13 访问控制符	153
7.14 接口	153
7.14.1 接口的声明	153
7.14.2 实现接口	154
7.14.3 扩展接口和接口的多重继承	156
7.14.4 关于菱形继承	157
7.14.5 使用其他的类实现的方法	158
7.15 Java 开发工具	160
习题	160
第 8 章 例外处理	163
8.1 Java 中的 Throwable 类及其子类	164
8.2 try-catch-finally 组合语句	165
8.3 throw 和 throws 语句	169
8.4 何时使用例外类	171
8.5 设计自己的例外类	172
8.6 例外的三大类	174
8.7 Java 例外处理机制与 C 语言处理比较	176
8.8 Java 例外处理机制的优点	176
习题	177
第 9 章 创建图形用户界面	178
9.1 如何设计用户界面的组件	178
9.1.1 图形界面的结构	178
9.1.2 最常用的 Component 类的方法	179
9.1.3 常用的组件和使用方法	179
9.2 Java 的布局管理	186
9.2.1 FlowLayout 类	186
9.2.2 BorderLayout 类	187
9.2.3 GridLayout 类	188
9.2.4 Insets 类	189
9.2.5 GradLayout 类	189
9.3 容器类	191
9.3.1 Window 类	192
9.3.2 Frame 类	192
9.3.3 Dialog 类和 FileDialog 类	193
9.4 菜单	193
9.4.1 下拉式菜单	194
9.4.2 弹出式菜单	195
9.5 action()方法	196

9.6 响应事件	203
9.6.1 事件代表模型	203
9.6.2 事件驱动程序设计	204
9.6.3 如何编写有菜单的 GUI 事件驱动程序	211
9.6.4 几个组件的事件处理	213
习题	215
第 10 章 多线程	216
10.1 线程的概念	216
10.1.1 多线程	216
10.1.2 线程与进程	216
10.1.3 多线程举例	216
10.2 线程的声明	218
10.3 线程的生命周期	219
10.3.1 线程的状态	219
10.3.2 线程状态的转换方法	221
10.4 Thread 类中的方法	223
10.4.1 Thread 类中的构造方法	223
10.4.2 实现线程行为的方法	224
10.4.3 改变线程状态的方法	224
10.4.4 其他方法	224
10.5 创建线程	225
10.5.1 扩展 Thread 类	225
10.5.2 使用 Runnable 接口	227
10.5.3 用 Runnable 接口创建线程的两种编程形式	227
10.5.4 创建线程的两种编程语句	229
10.5.5 多线程编程几个步骤	229
10.5.6 Daemon 监护线程	230
10.6 线程组	230
10.7 多线程同步	231
10.7.1 同步机制引入	231
10.7.2 实现同步的方法	232
10.7.3 如何使用 wait()方法、notify()方法和 notifyALL()方法	232
10.7.4 多线程同步机制的简化三原则	232
10.7.5 死锁问题	233
习题	233
第 11 章 Applet 技术基础	235
11.1 Applet 从建立到执行	235
11.2 HTML 文件中的 APPLET 标记	235
11.3 Applet 类和它的方法调用次序	238
11.4 用 Applet 画图形	240
11.5 Applet 的字型与颜色	241

11.5.1 字型	241
11.5.2 颜色	242
11.6 Applet 的图片处理	243
11.6.1 载入图片	243
11.6.2 画出图片	244
11.7 HTML 简介	244
11.8 举例——跳动的文字	245
习题	260
第 12 章 音乐和动画	261
12.1 声音和音乐	261
12.2 动画	264
12.2.1 重写 update() 方法	267
12.2.2 双缓冲技术	268
12.3 跟踪图像下载	268
12.3.1 MediaTracker 类	269
12.3.2 接口 ImageObserver	273
习题	275
第 13 章 Java 高级编程	277
13.1 Java 的输入/输出	277
13.2 Java 流类的层次结构	278
13.3 各种常用流类的使用方法	280
13.3.1 InputStream 和 OutputStream 抽象类	280
13.3.2 Reader 和 Writer 抽象类	281
13.3.3 FileInputStream 类和 FileOutputStream 类	282
13.3.4 FileReader 类和 FileWriter 类	282
13.3.5 ByteArrayInputStream 类和 ByteArrayOutputStream 类	283
13.3.6 CharArrayReader 类和 CharArrayWriter 类	284
13.3.7 PrintStream 类和 PrintWriter 类	284
13.4 File 类	286
13.4.1 文件对话框 FileDialog	287
13.4.2 对象文件 I/O	288
13.4.3 应用举例——存储雇员文件对象	290
13.5 网络编程	293
13.5.1 socket 和 TCP 通信	293
13.5.2 UDP 通信	297
13.5.3 URL 通信	297
13.5.4 URL 通信和 socket 通信的区别	300
13.5.5 多客户机通信	300
13.6 Java2 的新图形界面类库(Swing)	301
13.6.1 Java2 的主要特性	301
13.6.2 新的图形界面类库(Swing)	301

13.6.3	Java 插件	303
13.7	Java2 的各种实例	303
13.7.1	JApplet 小应用	303
13.7.2	JButton 小应用	305
13.7.3	滑动条 JSlider 小应用	308
13.7.4	Java2 综合小应用	309
13.7.5	带滑动条的进程条	310
13.7.6	用 JPasswordField 组件输入口令	311
13.7.7	选项板卡片容器 JTabbedPane	313
13.7.8	JTable 表格	315
习题		317
第 14 章	杂类	318
14.1	Object 类	318
14.2	内嵌类	318
14.3	包装类	319
14.4	Math 类	320
14.5	Random 类	321
14.6	Number 抽象类	322
14.7	Date 类和 Calendar 类	322
14.8	BigInteger 类	323
14.9	BigDecimal 类	323
14.10	System 类	324
14.11	Local 类	326
14.12	压缩类	328
14.12.1	制作压缩包的步骤	328
14.12.2	压缩包举例	329
习题		331
第 15 章	实例剖析—运动会计分系统	332
15.1	问题描述	332
15.2	设计与分析	333
15.3	开发步骤	333
15.4	开发环境和使用方法	334
15.5	运动会计分系统的程序框图及说明	335
15.6	运动会计分系统 Java 应用程序部分源代码	335
15.7	运行结果界面	340
习题		347
附录 A	UTF 码格式	349
附录 B	咖啡罐(JAR)	350
附录 C	如何使用 Native 程序	351
附录 D	远程方法调用	354
参考文献		357

第 1 章 Java 和 Web

1.1 计算机和万维网

今天,计算机的应用几乎涉及到各个领域:生活、办公、医药卫生、航天航空、金融、证券等等,几乎无所不在。20 世纪 90 年代,万维网(World-Wide Web—WWW)使计算机真正进入到我们每个人的日常生活。在因特网(Internet)上,你可以寄信,聊天,下棋,打电话,发短信,看电影,购物,订机票,找对象,玩游戏,查信息,做买卖,读小说,看当天或最新的报纸,发表看法和文章等等。目前,几乎 40% 的商业活动已转移到网上。电话、传真机、电视机、计算机将慢慢地消失,因为网络将会替你完成这些事件,你所需要的只是一台网络终端(Network Computer)。

也许,有人将来会这样撰写计算机的历史,计算机在 20 世纪出现,在 21 世纪消失在网络中。

万维网的雏形最初是瑞士高能物理研究中心实验室的 Time Berners-Lee 在 1989 年提出来的,HTML(Hyper Markup Language)语言也是他同时提出的。到了 1990 年 10 月,他的提议获得了批准,命名为“World-Wide Web”(万维网)。这个网络是基于因特网的。也就是说,万维网包括两个部分:浏览器和服务。浏览器向服务器提出请求,服务器为浏览器提供相关的信息。设计者 Time Berners-Lee 万万没有想到万维网会如此遍及全球,从当时的统计资料表明,万维网网点(服务器)个数是每 3~6 个月翻一番。信息检索服务站每天都有几千万至上亿人次用它们查找信息。至于每天在因特网上发送的 Email,用 ftp 传送的文件人次就难以用数字统计了。万维网以迅雷不及掩耳之势向全球每个角落扩张。Java 应运而生^①。“Write Once,Run Anywhere”使得 Java 魅力无限,用 Java 开发出来的软件可以不用修改或重新编译而直接用于任何计算机上,不管是 PC 机还是 UNIX 工作站,也不管它使用什么操作系统或者是哪个版本的操作系统,Java 程序都能安全地运行。

在 Java 之前,万维网上的网页基本是静态的,一个网点不能执行另一个网点上的程序,因为这两个网点可能用的是不同的计算机、不同的操作系统,或者是不同版本的软件。即使这些全一样,也还有个关键的安全性问题。Java 的出现迅速改变了这种状况。Java 的小应用(Applet)可以被下载到任何用户计算机上安全运行,这使得用户与网页的互动对话、动画展示等成为现实。用 Java 做成的动画优美流畅,音乐美妙动听,倾倒了无数用户。Java 为万维网带来了生命和一场真正意义上的技术革命。人们预言 Java 还将带来更深刻的革命。首先得到改变的是使用的软件方式。人们不用再去购买专用软件,然后没完没了地更新,只要在需要时访问专门网页,交付使用费,就能使用最新版本的软件。这样的网络将面向用户,给计算机“减肥”(减少 fatware);而不是像现在的微机那样面向一个大公司,经常要你去购买新的版本软件。

① 有关“Java 来历”,在下面章节将详细阐述。

1.2 程序设计语言和 Java

Java 是一种彻底的面向对象通用程序设计语言。Java 不但适用于网上程序设计,也适用于一般大规模软件工程项目。

程序设计语言通常划分为三类:机器语言、汇编语言和高级语言。机器语言仅是 CPU 懂得的程序设计语言。不同的 CPU 有它自己的机器语言,比如 Intel Pentium 和 Motorola PowerPc就有不同的机器语言。机器语言指令是低层二进制代码,这样我们必须提供庞大的机器指令才能完成一个简单的任务。比如,求 20 个数的平均值大约需要 15 条指令,十分繁琐。人们很快发现,这种机器语言实在难写又难读。于是就产生了第二类汇编语言。该语言替代了机器语言,成为比机器语言“高层”的符号语言。

比如计算机认得的 0011 0001 0010

如果 0011 代表加法,它就将 0010 单元中的数据加到 0001 单元中,即汇编语言替换成

ADD A,B

可以看到,汇编语言的每一条命令恰好对应一条机器语言。在 20 世纪 50 年代,当计算机速度太慢、内存太小之时,汇编语言起了很大作用。与此同时,计算机科学家们开始追求计算机语言的表达力,与机器无关性、结构化、模块化、信息封装和隐藏,以及更接近于我们日常用语、思维和数学公式,于是就产生了高级程序设计语言。比如 FORTRAN 语言(FORmula TRANslator),表达数学公式或数学方程,如:

$$Y = (X - 2) / (X - 1) + 2 * X * (X - 3)$$

COBOL (Common Business Oriented Language)是处理事务数据应用的高级程序设计语言。FORTRAN 产生于 1950 年末,而 COBOL 在 1960 年初出现。BASIC (Beginners Allpurpose Symbolic Instructional Code)是专为学生开发的早期语言。另一个高级语言 Pascal 是学术界的语言。C 语言是在 1970 年由 AT&T 贝尔实验室开发,C++ 语言是在 1980 年初开发的面向对象程序设计语言,它保留了 C 并且增加面向对象程序设计基本要素,成为混合式面向对象程序设计语言。

其实,世界上第一个高级语言并不是 FORTRAN,而是 K. Zuse 在 1944 年设计的 Plankalkul。该语言有变量、过程和参数,甚至强调了我们在 20 世纪 70 年代才开始重视的数据与程序结构,但不为人知。

从机器语言到汇编语言到过程语言(如 FORTRAN, Algol 60, PL/1, COBOL, BASIC, Pascal, C),到函数语言(如 Lisp, ML),到逻辑语言(如 Prolog),到面向对象语言(如 C++, Eiffel, Smalltalk-80, Ada),至今世界上已有 2000~4000 种程序设计语言。长期以来,在大学讲台上,经常面临着一个矛盾:是用 C 还是用 Pascal。Pascal 设计得好,但是太简单,工业界不用;C 最实用,但是用 C 写出来的程序实在不怎么样。我们知道,不好的语言带来了不好的编程习惯。Java 正是吸取了 50 年来程序设计语言研究的各种新思想以及正反面经验,形成一种好的程序设计风格的语言。好的程序设计风格,才有可能造就好的程序员。

Java 是有史以来第一个被工业界和学术界同时认可的语言。它从 1990 年开始至今不过 10 多年的时间,就能够拥有如此多的用户并受到这样的好评,确实值得从它的发展过程中探索其成功的原因。

Java 是 Sun 公司的产品, Sun 一直以经营工作站闻名的。但后来 PC 机的性能价格比并不比 Sun 工作站差, 故对 Sun 工作站构成了巨大压力。于是 Sun 公司成立了由 James Gosling 负责的一个名为 Green 的小组, 其目标是开发一系列诸如烤面包机、有线电视控制盒这样的消费产品, 以减轻来自 PC 的压力。开始, 他们用 C 语言编程, 但很快就发现由于 C 的编译过程与硬件密切相关性, 导致了一旦硬件系统更新, 用 C 的软件固化的芯片必须重新修改, 而电子类产品的各类芯片却在不断更新, 这类产品的软件可靠性要求又极高, 软件的一丁点故障就可能导致整个设备的替换, 这当然是商家所不愿看到的。这样 James Gosling 就力图设计一种独立于硬件平台的计算机语言解决这个问题。他也考虑过 C++ 语言是当前最广泛使用的语言, 但最终还是设计了一种以 C++ 为基础的语言, 尽量保留了 C++ 的语言的语素、关键字等, 融合了 C 和 C++ 等传统程序语言的优点, 形成了这套与众不同的面向对象通用程序设计语言。

但是, 由于商业上的种种原因, 当时这些电子产品没能够推向市场, Java 也差点夭折。

1993 年, 因特网的 WWW 由字符界面迅速发展到了图形界面, 极大地加快了因特网的发展速度。

1994 年, Sun 公司的元老之一 Bill Joe 参加了 Green 小组, 他经过认真思索, 决定将 Java 用在因特网的 WWW 开发中, 并取得了设计上成功。此时, Bill Joe 又做出了通过因特网让世界上无数软件设计者免费使用 Java 的决定! 于是, Java 就被越来越多的用户使用, 受到越来越多的重视。同时, 他们用 Java 开发了一个 Web 浏览器(Hot Java), 通过这个完全用 Java 语言设计的浏览器在因特网上展示了 Java 的风采, 这才真正有效地将 Java 推向全世界。

Java 的原名叫 Oak(橡树)。但申请注册商标时, 律师发现 Oak 已经有人用了。所以必须重新命名。他们想了许多名字, 比如 DNA, SILK, RUBY……其中也有 Java, 提议者是在喝一杯 Java 咖啡时无意之中想到的, 其他名字后来都被淘汰掉了。在 Java 文档中, 人们常常可以见到一杯冒着热气的咖啡。在英文 Java 参考书中, 几乎每一本都用一杯热咖啡图案来点缀每一页的页码。

Java 通常也称之为 Web 程序设计语言, 因为它在 Web 浏览器上运行的是用 Java 编写的 Applet 小应用程序, 这就说, 你需要 Web 浏览器执行 Java 小应用。Java 正在接受这种独一无二的特征, 即 Java 允许在因特网上动态地灵活地优雅地传播信息, 并且这个特征使 Java 更具无限魅力。当然, 我们讲解 Java 不限于只编写小应用上, 也讲述 Java 应用程序(Java Application)。Java 应用程序是不需要 Web 浏览器即可单独运行的程序, 它与其他程序设计语言编写的程序是一样的, 所以, 本书不仅包括 Java 的小应用也包括 Java 的应用程序, 特别是后者, 它更能教会我们应用面向对象程序设计语言的基本要素来进行面向对象设计。

1.3 Java 与 C, C++

Java 是从 C++ 而来, 经过去粗取精, 升华提高, 脱胎换骨。对于变量声明、参数传递、操作符以及流控制等, Java 保留了与 C, C++ 相似的传统技术, 使得熟悉 C, C++ 的程序员能够很方便地进行编程。与此同时, Java 为了实现其简单性、鲁棒性^①和安全性也摒弃了 C 和 C++

① Java 在编译和运行程序时, 都要对可能出现的问题进行检查, 以避免程序中错误的发生。鲁棒性是指系统遇到意外冲击时, 能根据环境改变自身结构或行为的机制。它是一个动态的、进化的、质的概念。具体是通过垃圾自动收集和面向对象的例外处理技术来达到鲁棒性。