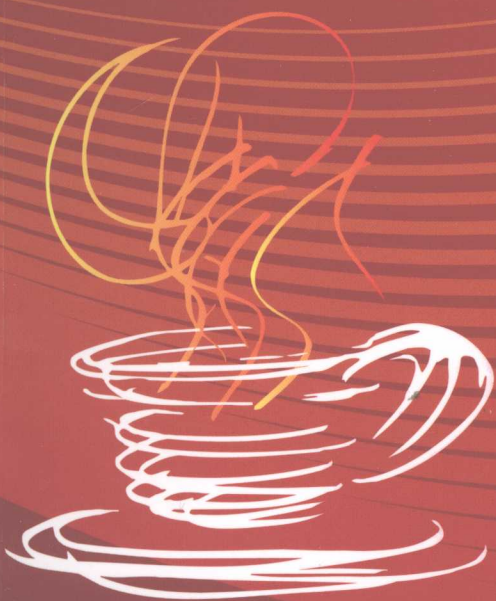




Java

语言基础教程

朱福喜 编著



- 循序渐进地阐述Java语言的基础知识与主要编程机制
- 大量的图表、浅显易懂的实例阐述使用Java语言的编程技巧
- 深入细致地将学习Java的难点、要点展示得淋漓尽致，易学易懂
- 通过精选大量Java认证和从业考试的试题，检验学习效果



清华大学出版社

内容简介



Java 语言基础教程

朱福喜 编著

清华大学出版社

北 京

内 容 简 介

本书从初学者的角度出发,利用通俗易懂的语言和大量生动典型的实例,由浅入深、循序渐进地介绍 Java 语言的基础知识与编程机制,并从面向对象程序设计的理论和方法中阐述使用 Java 语言的编程技巧。

全书列举了大量的图表和浅显易懂的实例,把学习 Java 的难点、要点展示的淋漓尽致、易学易懂。全书共分 9 章,包括:Java 概述、Java 语言基础、面向对象程序设计、流和文件、Applet、图形用户界面设计、异常处理、多线程、Java 网络编程。作者精心设计的综合实例——俄罗斯方块游戏,按照章节的内容安排和难易程度将实例分解并分布于各章节,章后配有自测题和练习题,为初学者的入门学习起到“切实(160 个实例)可行(226 个自测题+121 个练习题)”的指导性作用,并于书后提供自测题的参考答案。

全书结构严谨,兼有普及与提高的双重功能,可广泛适用于高等院校计算机及相关专业的 Java 语言教材,也适合作为软件开发人员及其他相关人员的参考书。

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 语言基础教程/朱福喜编著. —北京:清华大学出版社,2008.4

ISBN 978-7-302-17074-7

I. J… II. 朱… III. JAVA 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 021389 号

责任编辑:夏非彼 贾淑媛

责任校对:张楠

责任印制:王秀菊

出版发行:清华大学出版社

地址:北京清华大学学研大厦 A 座

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

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

印刷者:清华大学印刷厂

装订者:三河市金元印装有限公司

经销:全国新华书店

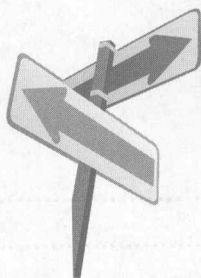
开本:185×260 印张:26.25 字数:639 千字

版次:2008 年 4 月第 1 版 印次:2008 年 4 月第 1 次印刷

印数:1~5000

定价:39.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:027309-01



前言

Java语言自1995年诞生以来，已走过10多年的光辉历程。早在Java诞生10周年之际，IT界就对Java的发展进行了回顾和展望，人们在分析Java对IT带来巨大影响的同时，也对Java的发展充满了无限的遐想和憧憬。Java在IT业的应用将一直保持强劲的增长势头，并将带动本世纪IT业的超常发展。

Java如此流行，它已成为IT从业人员的必备工具。

本书从Java的基本概念入手，介绍了主要编程机制、主要原理和方法。本书强调面向对象思维方法的训练，通过学习、设计及实现，使读者能掌握其方法、规范与理论，具有一定的面向对象分析、设计以及开发能力，为大型软件的研制、设计打下基础。

本书在内容和例题的选择上进行了精心的挑选，列举了大量的图表和程序实例，既要突出阐明原理和方法，又要保证一定的实用性，在难易程度上遵循由浅入深、循序渐进的原则。在举例过程中不仅注意到让读者能从实例运用中举一反三，同时也注意尽可能地站在读者的角度，体会Java语言的精髓，并在此基础上，掌握Java语言的编程技巧。也可以直接根据这些源程序来快速编写Java程序，直接切入相关应用。

本书可广泛适用于计算机及相关专业的专科、本科生，作为学习面向对象语言和学习Java语言的教材，也适合软件开发人员及相关人员作为自学的参考书或培训教程。

在本书的编写过程中，得到了清华大学出版社编辑的大力支持，在此谨向她们表示衷心感谢。

由于作者水平有限，书中难免存在疏漏和错误之处，恳请专家和广大读者批评指正。在学习过程中，如遇到疑难问题，可以通过以下方式与我们联系：booksaga@126.com，也可以登录图格新知网站<http://www.booksaga.com>留言，我们将在第一时间给予答复！

朱福喜

2007年10月于武昌珞珈山

目 录

第1章 Java概述	1
1.1 Java十周年	1
1.2 面向对象程序设计概述	4
1.2.1 对象	6
1.2.2 类	6
1.2.3 消息	7
1.2.4 继承	7
1.2.5 多态性	8
1.2.6 抽象	8
1.2.7 封装	9
1.3 Java语言的特色	10
1.4 JDK的下载、安装与环境配置	11
1.5 Java编译器与Java虚拟机	13
1.5.1 Java编译器	13
1.5.2 Java虚拟机	13
1.6 Java独立应用程序的基本结构	14
1.6.1 程序的注释	15
1.6.2 类的定义	15
1.6.3 方法的声明	16
1.6.4 程序的执行语句	16
1.6.5 一个应用程序从编辑到执行的完整过程	16
1.7 Java Applet程序的基本结构	19
自测题	21
练习题	22
第2章 Java语言基础	23
2.1 基本数据类型	23
2.1.1 简单类型	24
2.1.2 引用类型	24
2.1.3 常量	25
2.1.4 变量的声明和保存	26
2.2 数值类型的运算	28
2.2.1 数值类型的基本运算	28
2.2.2 位运算	28
2.3 数值类型的高级运算	29
2.3.1 数学函数运算及数学常量	29
2.3.2 快捷操作符和类型转换	30



2.4 逻辑和比较	33
2.4.1 逻辑操作符	33
2.4.2 比较操作符	34
2.4.3 instanceof操作符	34
2.5 条件执行语句	34
2.5.1 简单的if语句	35
2.5.2 if-else选择语句	35
2.5.3 嵌套的if-else-if语句	37
2.5.4 switch语句	38
2.6 循环控制语句	40
2.6.1 循环的概念	40
2.6.2 for循环	40
2.6.3 while循环	41
2.6.4 do-while循环	42
2.6.5 break和continue语句	43
2.7 数组	43
2.7.1 声明数组	43
2.7.2 初始化数组	44
2.7.3 访问数组的元素	44
2.7.4 数组作为方法的参数	45
2.8 字符串	46
2.8.1 字符串类String	46
2.8.2 StringBuffer类	51
2.8.3 通过输入字符串来输入各种基本类型	52
2.9 综合实例——俄罗斯方块游戏	53
2.9.1 游戏介绍	54
2.9.2 用数组表示方块的28种状态	54
2.9.3 位运算和位移运算在游戏中的使用	55
2.9.4 用for语句和break语句在游戏中的联合使用	55
2.9.5 用for语句和continue语句在游戏中的联合使用	56
自测题	57
练习题	61
第3章 面向对象程序设计	62
3.1 类与对象	63
3.1.1 类的定义	63
3.1.2 创建对象和构造方法	65
3.1.3 垃圾回收机制	69
3.2 类成员的定义	71
3.2.1 成员变量的定义	71
3.2.2 成员方法的定义	71
3.2.3 变量的作用域	76
3.3 类的访问与封装	77



3.3.1 访问成员变量与成员方法	78
3.3.2 成员的访问权限的控制	79
3.3.3 静态成员变量与静态成员方法	84
3.4 重载	88
3.4.1 重载成员方法	89
3.4.2 重载构造方法	89
3.5 继承与覆盖	91
3.5.1 类的继承层次	91
3.5.2 继承与构造方法	95
3.5.3 覆盖	97
3.5.4 抽象与终结	99
3.5.5 Java的基类Object	102
3.6 接口与多态	105
3.6.1 多重继承和接口	105
3.6.2 多态性	107
3.7 实现多态性下的计算	110
3.7.1 用抽象类实现多种形状面积的累加	110
3.7.2 用接口实现多种形状面积的累加	111
3.7.3 用一个Object数组实现多种形状面积的累加	113
3.8 包	114
3.8.1 Java平台定义的包	114
3.8.2 包的创建	115
3.8.3 包的使用	116
3.8.4 编译和生成包	116
3.9 综合实例	117
3.9.1 俄罗斯方块游戏中的类的继承关系	117
3.9.2 俄罗斯方块游戏中的ErsBox类	118
自测题	120
练习题	125
第4章 流和文件	127
4.1 流的基本概念	127
4.2 文件类及其操作	129
4.2.1 创建与删除文件	130
4.2.2 获取文件或目录属性	130
4.2.3 获取目录成员	132
4.2.4 过滤文件名	133
4.3 字节级输入输出类	135
4.3.1 文件输入流 (FileInputStream)	136
4.3.2 文件输出流 (FileOutputStream)	138
4.3.3 字节数组输入流 (ByteArrayInputStream)	139
4.3.4 字节数组输出流 (ByteArrayOutputStream)	140
4.3.5 回退输入字节流 (PushbackInputStream)	141



4.3.6 数据输出流 (DataOutputStream)	143
4.3.7 数据输入流 (DataInputStream)	145
4.3.8 缓存输出流 (BufferedOutputStream)	146
4.3.9 缓存输入流 (BufferedInputStream)	148
4.3.10 格式化输出流 (PrintStream)	150
4.4 字符级输入输出类	151
4.4.1 字符输入流 (Reader)	152
4.4.2 字符输出流 (Writer)	152
4.4.3 FileReader类	153
4.4.4 BufferedReader类	153
4.4.5 FileWriter类	155
4.4.6 BufferedWriter类	155
4.4.7 PrintWriter类	156
4.4.8 读取字符数组类 (CharArrayReader)	158
4.4.9 写字符数组类 (CharArrayWriter)	159
4.4.10 回退输入字符流 (PushbackReader)	160
4.5 流的标记分解类 (StreamTokenizer)	162
4.6 对象级输入输出类	164
4.6.1 Serializable接口	164
4.6.2 对象输出流 (ObjectOutputStream)	164
4.6.3 对象类输入流 (ObjectInputStream)	165
4.7 RandomAccessFile	167
4.8 系统I/O流	169
自测题	172
练习题	176
第5章 Applet	177
5.1 Applet的基本概念	177
5.1.1 Applet的构架	177
5.1.2 Applet类的实例变量	179
5.1.3 Applet类的成员方法	179
5.1.4 Applet的生命周期	180
5.2 Applet标记	182
5.2.1 Applet标记中的属性	182
5.2.2 利用标记向Applet传递参数	183
5.3 Applet的安全限制与文件读取方法	185
5.3.1 统一资源定位器URL	186
5.3.2 获取URL	187
5.3.3 使用URL载入数据	189
5.4 Applet环境与状态显示	191
5.4.1 Applet的状态显示	191
5.4.2 Applet的环境的获取和利用	193
5.5 Applet的绘图与控制	195



5.5.1 在Applet中利用Graphics类绘图	195
5.5.2 在Applet中显示与更新	199
5.5.3 在Applet中加入控制项	201
5.6 综合实例——交通灯控制	204
自测题	206
练习题	209
第6章 图形用户界面设计	211
6.1 抽象窗口工具包 (AWT) 简介	211
6.2 基本的窗口类	212
6.2.1 Container类	212
6.2.2 Window类	212
6.2.3 Panel类	213
6.2.4 Frame类	213
6.2.5 Dialog	213
6.2.6 简单窗口举例	214
6.3 简单的GUI构件类	215
6.3.1 Button (按钮) 类	215
6.3.2 TextField (文本域) 类	215
6.3.3 Label (标签) 类	215
6.3.4 TextArea (文本区) 类	215
6.3.5 List (列表) 类	216
6.4 简单的事件处理	217
6.4.1 动作事件响应的一般步骤	217
6.4.2 ActionListener接口	218
6.5 事件类型及其处理	219
6.5.1 事件源	220
6.5.2 事件监听接口	221
6.5.3 低级事件	222
6.5.4 高级事件	230
6.5.5 事件、事件源与监听接口	239
6.5.6 监听接口适配器	240
6.6 布局管理	241
6.6.1 FlowLayout	241
6.6.2 BorderLayout	242
6.6.3 GridLayout	243
6.6.4 CardLayout	245
6.6.5 GridBagLayout	247
6.7 菜单系统	250
6.8 对话框	253
6.8.1 Dialog类	253
6.8.2 FileDialog类	255
6.9 图形类与图形的绘制	258



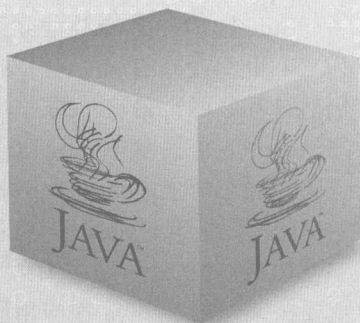
6.9.1 画线	258
6.9.2 画矩形	262
6.9.3 设置色彩和画其他图形	264
6.10 AWT与Swing	266
6.11 综合实例——俄罗斯方块游戏的界面实现	268
6.11.1 主类ErsBlocksGame	268
6.11.2 控制面板类ControlPanel	276
6.11.3 游戏显示窗口GameCanvas	281
自测题	283
练习题	286
第7章 异常处理	288
7.1 错误和异常的产生与处理	288
7.2 Java平台定义的异常	291
7.2.1 异常类的层次结构	291
7.2.2 Throwable 类	291
7.2.3 Error类和Exception类	292
7.2.4 隐式和显式异常	293
7.3 自定义的异常	295
7.4 抛出一个异常	298
7.4.1 抛出一个显式异常	298
7.4.2 处理异常的方法	299
7.5 捕获异常	303
7.5.1 try-catch错误处理	303
7.5.2 多异常的捕获和处理	306
7.5.3 finally子句	309
7.6 异常处理的嵌套	312
7.6.1 在try块中包含有try-catch语句	312
7.6.2 在捕获异常之后又出现了异常	313
7.6.3 在finally子句中包含有try-catch语句	314
7.7 俄罗斯方块游戏中的异常处理	315
7.7.1 在ErsBox类中克隆对象时捕获异常	316
7.7.2 在ErsBlock类中克隆对象时捕获异常	316
7.7.3 在ControlPanel类中克隆对象时捕获异常	316
自测题	317
练习题	320
第8章 多线程	322
8.1 线程的基本概念	322
8.1.1 线程类Thread	323
8.1.2 线程的状态	324
8.1.3 线程的优先级	325
8.2 线程的使用方法	325
8.2.1 通过继承Thread类创建线程	325



8.2.2 通过实现Runnable接口创建线程	327
8.3 多线程的使用方法	329
8.3.1 多线程的创建方法	329
8.3.2 判断线程是否存在	333
8.3.3 线程的结合 (join)	333
8.3.4 线程的优先级的获取和设置	335
8.4 多线程的同步处理	336
8.4.1 线程synchronized方法	337
8.4.2 Applet中使用线程	339
8.4.3 线程的等待与唤醒	341
8.5 综合实例——俄罗斯方块游戏中使用的线程	344
8.5.1 通过继承Thread类实现俄罗斯方块游戏	344
8.5.2 通过Runnable接口实现游戏类方块	348
自测题	350
练习题	354
第9章 Java网络编程	356
9.1 Java网络应用基础	356
9.1.1 IP地址	357
9.1.2 端口	358
9.1.3 套接字	358
9.1.4 数据包	359
9.1.5 Internet协议	359
9.2 基于URL网络应用	360
9.2.1 直接从URL读取内容	360
9.2.2 建立一个URL连接并从中读取内容	361
9.3 以Socket方式实现通信	362
9.3.1 客户端Socket	362
9.3.2 服务器端Socket	363
9.3.3 简单邮件系统	369
9.3.4 有关目录的客户/服务器系统	371
9.4 用Datagram方式实现通信	382
9.4.1 数据包和套接字	383
9.4.2 用Datagram实现客户/服务器模式	384
9.5 综合实例——俄罗斯方块双人对战网络游戏	388
9.5.1 游戏的实现概要	388
9.5.2 基于Socket的C/S模式的通信	389
9.5.3 对打游戏具体实现	391
9.5.4 游戏的运行	400
自测题	402
练习题	404
自测题答案	405
参考文献	409

第 1 章

Java 概述



自从 Java 语言产生以来，一直以它独特的魅力征服了整个 IT 行业。它不仅成为一门最为流行的计算机语言，而且还形成一种专门的技术，从而对 IT 行业产生巨大的推动和影响。

1.1 Java 十周年

Java 技术已出现十多年，Java 十周年之际，回首 Java 走过的历程，总结了 Java 十年来的十件大事、十大人物、十大产品、十大组织以及十大热点问题等，对 Java 进行了认真的回顾和思考。其中从 Java 十件大事，可以看到 Java 走过的光辉历程。

1. 1990~1994: Java 的起源

1990 年 12 月，Sun 的工程师 Patrick Naughton 启动了一个叫做 Stealth（秘密行动）的项目。随着 James Gosling 等人的加入，这个项目更名为 Green。其目标是使用 C++ 为嵌入式设备开发一种新的基础平台技术，James Gosling 本人负责开发一个 SGML（标准通用标记语言）编辑器，并决定开发一种新的编程语言。很快这种颇受同伴喜爱的小语言被命名为 Oak。但是，这个在技术上非常成功的产品，当时并未获得商业上的成功。后来赶上互联网发展的大潮，1994 年，Oak 被命名为 Java。

2. 1995: Java 问世

1995 年，Sun 正式对外公布了 Java，并且发布了 JDK 1.0。这种外形酷似 C++，却包含一颗类似 Smalltalk 的纯洁的面向对象之心的全新程序设计语言及其平台，刚一问世就成为软件产业的新宠儿。Java 当时仅仅被用来为网站制作一些动态应用，诸如动画图片之类，但这仍然引起了很多 Web 开发者们的注意，他们非常渴望有一种安全的语言，可以在静态的 HTML 网页上制作动画图片，这导致 Sun 公司最终把 Java 集成到 NetScape 浏览器中。同时因为它具有“只写一次，随处运行”的特性，引起了很多开发者的青睐，因为这样他们再也



不用为了使程序能够在不同型号的硬件上运行而耗费大量的时间来编译代码了。

Web 浏览器的出现也为 Java 的出现起到了很好的推动作用, 将 Java 和 Web 浏览器结合, 人们似乎看到了某种希望。Java 的出现为当时的软件产业带来了无限的遐想。

3. 1996: Java 大跃进, 崭露头角

Sun 在 1996 年首先成立了 JavaSoft 组织, 并在 1 月 23 日正式发布自己的 Java 1.0。作为 20 世纪业界出现的最重要的技术之一, Java 引起了编程世界的革命。直到现在, Java 仍然是互联网上最流行的语言。

在 Sun 正式发布 Java 1.0 之后, Java 这门新生的语言就拥有了自己的讲坛——JavaOne, 各大知名公司也纷纷向 Sun 申请 Java 的许可证。一时间, NetScape、惠普、IBM、Oracle、Sybase, 包括当时刚推出 Windows 95 的微软都是 Java 的追随者。

现在, J2EE 已经成为中大型企业级应用的标准, 成为承接数据库和 Web 之间的一个重要桥梁。最终 Java 在应用服务器市场获得了难以取代的地位, 也确定了 J2EE 的发展方向, 并且仍将延续下去。

4. 1997~2001: 微软与 Sun 的 Java 之战

Java 诞生的 1995 年, 正是微软在软件产业地位达到巅峰之时, 但比尔·盖茨仍然敏锐地注意到 Java 的重要性, 对 Java 给予了这样的评价: “Java 是很长时间以来最优秀的程序设计语言”。并且, 微软于 1996 年 3 月申请并获得了 Java 许可证。微软对于 Java 的热情态度在当时大大提高了人们对 Java 的兴趣和信心, 但也有不少人担心微软会依靠自己强大的影响力在标准之外另立标准, 从而破坏 Java 的纯洁性。

果然, 从 1997 年发布 Visual J++ 的第一个版本开始, 微软就开始在 Java 中掺入自己的私有扩展。此举引起 Sun 的高度重视。1997 年 10 月, Sun 向美国加州地方法院起诉微软公司违反两公司关于使用 Java 技术所签订的合同, 指控微软公司在自己的 Java 产品中做了“不恰当的修改”, 违反了合同中承诺向用户提供 Java 兼容产品的条款。这一官司旷日持久, 直到 2001 年 1 月双方才达成和解, 微软将继续提供采用 Sun 开发的 Java 技术的现有产品 (包括测试版)。到了 2001 年 7 月, 微软公布新版的 Windows XP 不再支持 Sun 的 JVM (Java 虚拟机), 并且推出了 .NET 平台与 Java 分庭抗礼。

这一场官司对 Java 世界产生了深远的影响。如果没有这一场官司, 人们将不得不面对被分裂的 Java 世界。

5. 1998: Java 2 平台发布

1998 年, Java 2 平台正式发布。经过了三年时间的发展, Sun 终于知道 Java 适合干什么了。根据官方的文档, Java 2 是 Sun 意识到 “one size doesn't fit all” 之后, 把最初的 Java 技术打包成三个版本: J2ME、J2SE、J2EE。而在 Java 2 的三大版本中, 除了 J2EE 得到了长足发展和广泛使用之外, J2ME 也在手机市场上取得了遍地开花的结果。无论今天看来当时的 Java 2 有多么不成熟, 至少经过市场和时间的检验, Java 2 规划出来的三大方向把 Java 技术指向了光明大道。

6. 1998: JCP 成立, Java 开源社团蓬勃发展

1998 年, JCP 组织成立, 并且开始把握 Java 的发展方向。JCP 组织的开放性, 不但使得所有对 Java 感兴趣的商业公司可以参与 Java 的发展, 更重要的是 JCP 允许个人、非盈利组织、学校等加入, 这就给 Java 带来了巨大的活力。随之兴起的 Java 开放源代码运动的最大贡献是实现和鼓励了知识共享, 在众多热情的开放源代码程序员们的努力和分享下, 很多原先只被商业公司掌握的技术、思想和产品可以被所有需要它的开发人员免费或者以较低的价格获得使用权, 并通过开放源代码更容易地获得反馈和改进意见, 从而进一步演化发展。

20 世纪 90 年代可以说是 IT 产业的一个黄金时代。信息时代的兴起对 IT 人员, 特别是软件人员有着巨大的需求。而软件开发又是一种类似艺术创作的脑力活动, 和所有的艺术家、作家们一样, 在作品打上自己的印记并流传是每一个创作人员的梦想。今天, 开放源代码运动已经不仅仅是一些天才程序员的游乐园地, 而是发展成为一项开放源代码软件产业。

7. 1998: WebLogic 打开 J2EE 的魔匣

Java 语言的出现使得互联网络有了良好的交互性能, 但这些很“酷”的技术仅被人们认为是一些小花招, 它还无法消除企业级用户对它的怀疑。1998 年, BEA 公司宣布收购 WebLogic 公司, 并接着推出由 Sun 公司第一个授权使用 J2EE 许可证的 WebLogic Server 应用服务器, 这个 Java 版的 AppServer 刚推出就引起业界的极大兴趣。WebLogic Server 以其对标准的支持、强悍的运算能力和安全的架构设计等特性也很快征服了那些怀疑 J2EE 应用的人们。推出市场后不到一年, WebLogic Server 就成为业内排位第一 Java 应用服务器。

8. 2002~2004: Sun 与微软的法律碰撞最终以喜剧收场

2003 年 4 月 2 日, Sun 与微软达成 16 亿美元的法律和解。如果不是晚了一天, 许多人会以为这是一个在 4 月 1 日愚人节开的玩笑。尽管当时所有人都像是看到“太阳从西边出来了”那样张大了嘴巴, 但这的确是事实。根据两家公司达成的版权协议, 双方会为采用对方的技术而支付专利费用, 微软向 Sun 提前支付 3.5 亿美元使用费, Sun 则承诺, 如果 Sun 集成微软的某些技术, 也会向微软付款。

9. 2000~2004: JBoss 和 Eclipse——Java 开源软件的王者

目前最有影响力的 Java 开源软件项目, 要数 JBoss 和 Eclipse。前者是目前最优秀、应用最为广泛的企业级开源 J2EE 应用服务器, 后者是功能完全可以替代商业产品的 Java IDE (集成开发环境)。二者的覆盖功能之全、支持工具之广、子项目之多, 几乎可以仅凭借它俩来完成企业应用的从开发构建到部署实施的全过程, 而软件开发者和客户也都可以在最大程度上享受高质量、高可靠 Java 开源软件所带来的低成本优势。

10. 2004: Java 5.0 闪亮登场

2004 年 9 月 30 日, 代号为 Tiger, 研发历时近三年的 J2SE 5.0 发布正式版本, 这是 Java



平台历年来发布版本中改动面波及最大的一次。“Ease of development”是 J2SE 5.0 核心主题，这一点着重体现于语言特性上的改进，在很大程度上简化了开发人员日常的编程任务，同时对构筑于 J2SE 5.0 之上的 J2EE 5.0 产生了巨大影响。J2SE 5.0 发布后，Sun 还会进一步规划 J2SE 的未来蓝图。

与 Java 十件大事同时广为人们关注的是 Java 十大产品、十大人物、十大组织。作为一般了解，下面仅在图 1.1、图 1.2、图 1.3 给出它们的名称。

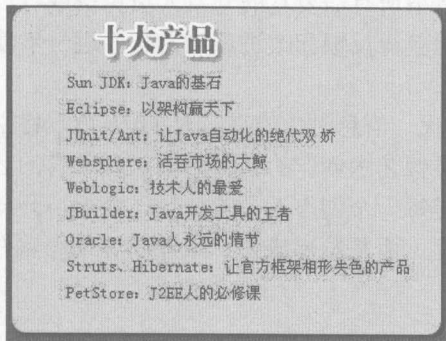


图 1.1 Java 的十大产品

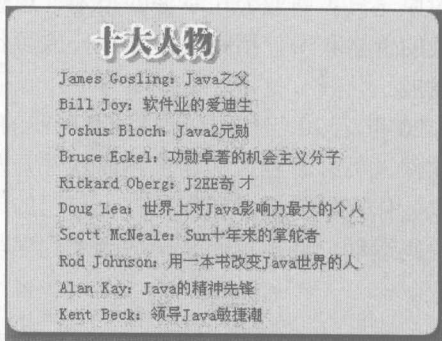


图 1.2 Java 的十大人物

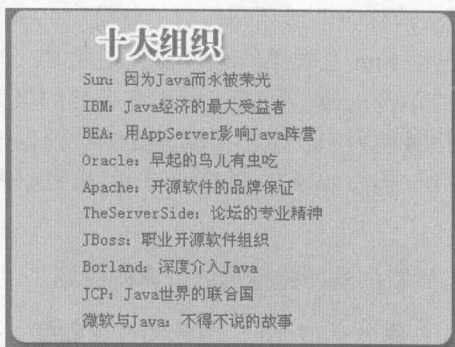


图 1.3 Java 的十大组织

1.2 面向对象程序设计概述

Java 以纯面向对象的程序设计语言著称，所以有必要讨论一下面向对象的基本理论和基本技术。

面向对象技术代表一种自然地观察、表述、处理问题的方法和程序设计思路。面向过程的程序设计是以具体的解题过程为研究和实现的主体，而面向对象的设计则是以需要解决的问题中所涉及到的各种对象为主要矛盾，关心的是对象及其相互之间的联系，问题求解过程力求符合人们日常自然的思维习惯，降低和分解问题的难度和复杂性，提高了整个求解过程的可控性、可监测性和可维护性，从而以较小的代价获得较高的效率和较满意的结果。

面向对象的程序设计是以对象为中心,按照对象及其联系来构造实现软件单位的程序设计。面向对象的程序设计将在面向对象的问题求解所形成的对象模型基础之上,选择一种面向对象的高级语言来具体实现这种模型。相对于传统的面向过程的程序设计方法,面向对象的程序设计具有更好的可重用性、可扩展性和可管理性,其优点具体主要体现在如下几个方面。

1. 封装性

对象的封装性彻底消除了传统结构方法中数据与操作分离所带来的种种问题,降低了维护数据与操作之间的相容性的负担。而且把对象的私有数据和公共数据分离开,保护了私有数据,减少了模块间可能产生的干扰,达到降低程序复杂性、提高可控性的目的,提高了程序的可重用性和可维护性。

2. 自治性

对象作为独立的整体具有良好的自治性,即它可以通过自身定义的操作来管理自己。一个对象的操作可以完成两类功能:一是修改自身的状态;二是向外界发布消息。当一个对象想要影响其他对象时,需要调用那个对象的方法,而不是直接去改变那个对象。对象的这种自治性能够使得所有修改对象的操作都可以以对象自身所具有的一种行为的形式存在于对象整体之中,从而维护了对象的完整性,有利于对象在不同环境下的重用、扩充和维护。

3. 安全性

对象具有通过一定的接口和相应的消息机制与外界相联系的特性,并与对象的封装性结合在一起,较好地实现了信息隐藏。这样使得对象成为一只使用方便的“黑匣子”,其中隐藏了私有数据和内部运行机制。使用对象时只需要了解其接口提供的功能和操作,而不必了解对象内部的数据描述和具体的功能实现。

4. 扩展性

继承是面向对象的另一个重要特性,通过继承可以很方便地实现应用的扩展和已有代码的重复使用,在保证质量的前提下提高开发效率,也使得面向对象的开发方法与软件工程的一个新方法——快速原型法能够很好地结合在一起,形成一种更有效、更实用的软件开发技术。

综上所述,面向对象程序设计是将数据及数据的操作封装在一起,成为一个不可分割的整体,同时将具有相同特征的对象抽象成为一种新的数据类型——类。类是构造软件系统的最小单位,一个程序结构就是一个类的集合以及各类之间以继承关系联系起来的结构。围绕着类还有一些相关的概念:从类的定义创建实例就产生了对象;对象间通过消息进行互动;类与类的从属关系产生了父类与子类关系,父类和子类关系具有继承性;子类有变异现象产生了多态性;如何创建类需要抽象手段;创建类之后就具有封装性等。下面就这些基本概念进行阐述。



1.2.1 对象

关于对象的概念是理解面向对象技术的关键。对象在不同的上下文中可能有不同的含义。在面向对象的程序设计方法中，对象是一些相关的变量和方法的软件集，是可以保存状态（信息）和一组操作（行为）的整体。软件对象经常用于模仿现实世界中的一些对象，比如桌子、电视、自行车等。现实世界中的对象有两个共同特征：状态和行为。比如，自行车的状态有车的样式、颜色、当前档位、两个轮子的大小等，其行为有刹车、加速、减速以及改变档位等。

软件对象的状态和行为可以分别用对象中的变量和方法来表达，图 1.4 (a) 是一个软件对象的直观表示。例如，构造现实世界中的自行车的软件对象要有指示自行车的当前状态的变量：速度为 20km/h，它的当前档位为第 3 档。这些变量就是实例变量，表示特定自行车对象的状态。在面向对象技术中，特殊的对象称为实例。

除了变量之外，描述自行车的对象还有用于刹车、改变档位的方法。有了一个实例对象之后，这些方法就是实例方法，它们能够设置或者改变特定自行车实例的状态。自行车的状态和行为的描述见图 1.4 (b)。

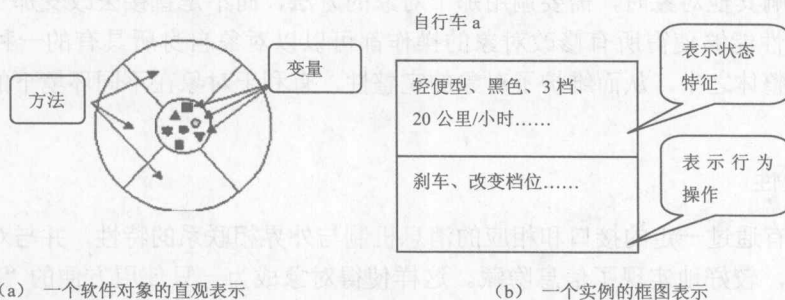


图 1.4 对象的表示

1.2.2 类

在现实世界中，许多对象属于同一类型，俗话说：“物以类聚”。在软件世界中，类实际上是对某种具有共同特征类型的“一类”对象的定义，即类定义了一类对象的类型。属于该类的所有对象都具有相同的变量和方法。

类是同一种对象的集合的抽象，即同一类对象的变量和方法的原型。在面向对象软件中，可以让许多对象共有的一些相同特性形成类，然后可以利用相同类型的对象模型创建一个类，表达同一种对象的共同的属性和特点，所以类是对象的软件模型，是一个抽象定义、一个模板。因此，也有人把类称为抽象数据类型。

在创建汽车类的时候，需要定义一些实例变量来包括当前档位、当前速度等，同时这个类也需要为操作这些变量提供方法定义和方法实现，允许开车者减速、刹车以及加速等。

当创建了汽车类以后，就可以用这个类创建任意多个汽车对象。当创建了一个类的对象后，系统将为这个对象的实例变量分配内存。这样，每个对象的实例变量可存放各自不同的状态值。类与对象的关系如图 1.5 所示。