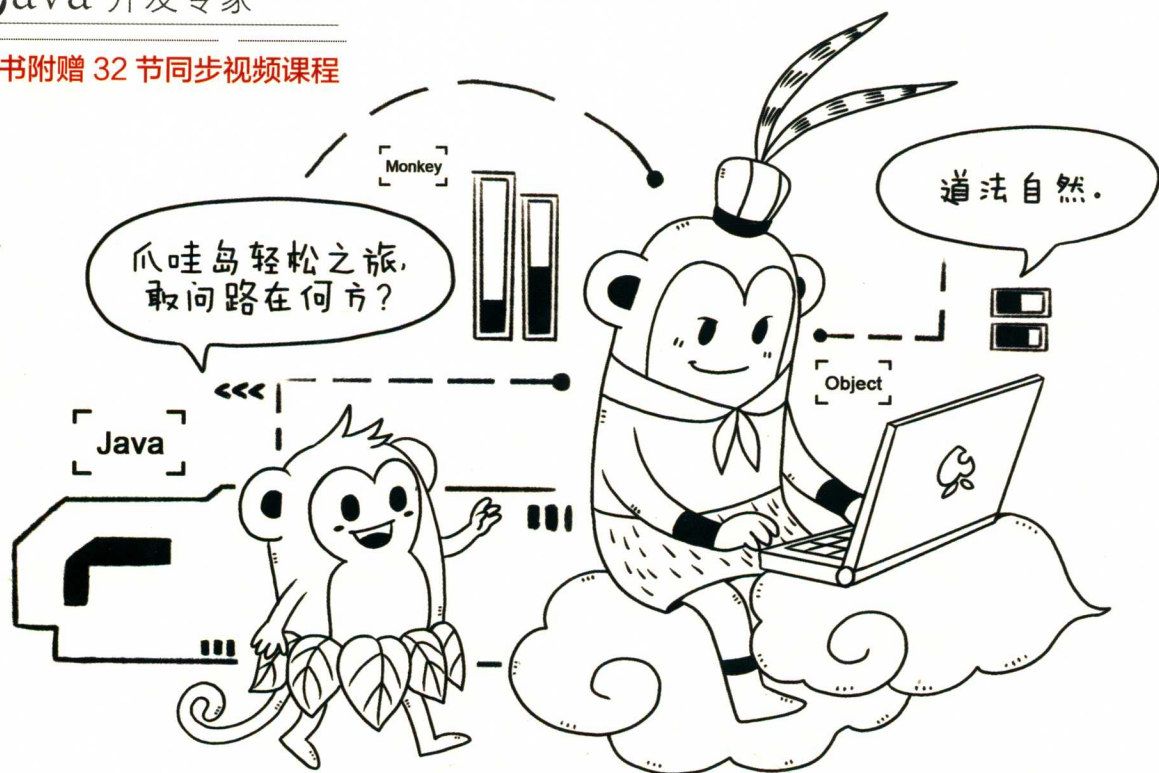


Java 开发专家

随书附赠 32 节同步视频课程



大话 Java

程序设计从入门到精通

孙卫琴 编著

深入浅出，妙趣横生，
Java轻松进阶无难事。



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



大话 Java

程序设计从入门到精通

孙卫琴 编著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

· 内 容 简 介 ·

本书采用由浅入深、与实际应用紧密结合的方式,利用大量典型生动的范例,详细讲解了 Java 编程的各种基本技术。本书的范例全部基于最新的 JDK9 版本。本书内容包括:创建和运行 Java 程序的基本方法、Java 语言的基本语法、数据类型和变量、操作符、流程控制、继承、Java 语言中的修饰符、接口、异常处理、对象的生命周期、内部类、多线程、数组、集合、输入输出、图形用户界面和访问数据库。本书还介绍了 JDK9 的一些新特性,包括 JShell 命令及 Java 类库的模块化等。

本书别出心裁,引入了中国传统文化中家喻户晓的《西游记》中的人物孙悟空,以他学习 Java 语言为主线,以模拟《西游记》中的种种传奇故事及实现各种有趣的实际应用为案例,带领读者逐步领略 Java 语言的种种神通妙用,大大增加了书的趣味性。

书中实例源文件请到 JavaThinker.net 网站上下载,网址为: <http://www.javathinker.org/funnyjava.jsp>。

本书主要面向所有的 Java 初学者,还可作为高校的 Java 教材,以及企业 Java 培训教材,也可作为 Oracle 公司 OCJP 认证的辅导教材。

本书配套光盘包含书中案例源代码、视频教程和 PPT 讲义。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

大话 Java:程序设计从入门到精通 / 孙卫琴编著. -- 北京:电子工业出版社,2018.9

ISBN 978-7-121-34930-0

I. ①大… II. ①孙… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 196835 号

责任编辑:张艳芳 文字编辑:张琳 特约编辑:刘红涛

印 刷:湖北画中画印刷有限公司

装 订:湖北画中画印刷有限公司

出版发行:电子工业出版社

北京海淀区万寿路 173 信箱 邮编:100036

开 本:787×1092 1/16 印张:25 字数:637.6 千字

版 次:2018 年 9 月第 1 版

印 次:2018 年 9 月第 1 次印刷

定 价:79.90 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254161~88254167 转 1897。

Java 语言是目前 IT 领域里的主流编程语言。面向对象的 Java 语言具备一次编程、在任何平台中均可运行的跨平台特性，在需要支持多种操作系统和硬件平台的场合，Java 是首选的解决方案。

Java 语言非常安全和健壮。Java 致力于检查程序在编译和运行时的错误，奉行“错误发现和纠正得越早，造成的损失就越小”的原则，可谓防患于未然。Java 还支持自动内存管理，这不但减轻了程序员的许多负担，也减少了程序员错误释放内存的机会。

Java 语言自 1996 年诞生以来，其开源的精神吸引了世界各地的 IT 精英们不遗余力地为它添砖加瓦。在如今的 Java 领域，各种新技术、新工具层出不穷，一方面，每一种技术都会不停地升级换代，另一方面，还会不断涌现出新的技术和工具。Java 世界就像小时候玩的万花筒，尽管实质上只是由几个普通的玻璃碎片组成的，但只要轻轻一摇，就会变化出千万种缤纷的图案。Java 世界如此变化多端，很容易让初学 Java 的人有无从下手的感觉。

常常会有 Java 新手问我类似这样的问题：“我学 Java 已经有一段时间了，现在只能编写一点简单的程序，要学的东西实在太多了，我整天学都学不完，很迷茫，不知道该如何有针对性地去学，才能早日成为一名功底深厚的 Java 程序员。”

确实，对于初学者，一开始就陷入包罗万象的 Java 技术的汪洋大海中，难以把握技术的核心思想，以及知识的深浅和主次，学习起来会比较吃力。

为了满足广大 Java 初学者的要求，本人在创作了十多本 Java 领域里的畅销书籍后，融合近二十年来的软件开发、教学和写作经验，用轻松诙谐的笔调，创作了《大话 Java》这本书。本人在动手写每一个知识点时，有三个问题时时在脑袋里激发自己的灵感：“怎样写才能让读者一看就懂？怎样写才能增添书的趣味性，激发读者的学习兴趣？设计怎样的范例才能让读者迅速掌握实际运用的技能？”这三个问题激励着我精益求精地设计书中的范例，推敲书中的措辞，精炼书的结构。

古人云：“授人以鱼，不如授人以渔。”在一本 Java 书中，泛泛而谈地罗列所有的技术，如同授人以鱼，而以抛砖引玉的方式引导读者把握 Java 编程的核心思想，并且掌握灵活运用技术进行编程的技能，则如授人以渔。本书致力于引领 Java 初学者们在 Java 领域里乘风破浪，游刃有余地“织网渔猎”。

本书的组织结构和主要内容

本书别出心裁，引入了中国传统文化中家喻户晓的《西游记》人物孙悟空，以他学习 Java 语言为主线，以模拟《西游记》中的种种传奇故事及实现各种有趣的实际应用为案例，带领读者逐步领略 Java 语言的种种神通妙用，大大增加了书的趣味性。

本书的每一章都按照提出问题和解决问题的结构来安排内容，并且提供了编程实战练习，引导读者由被动阅读转为主动阅读，从而使读者既能深刻地领悟各种 Java 知识的用途，又能提高运用特定技术来解决实际问题的能力。

本书主要内容包括：创建和运行 Java 程序的基本方法、Java 语言的基本语法、数

• Foreword •

据类型和变量、操作符、流程控制、继承、Java 语言中的修饰符、接口、异常处理、对象的生命周期、内部类、多线程、数组、集合、输入输出、图形用户界面和访问数据库。本书还介绍了 JDK9 的一些新特性，包括 JShell 命令及 Java 类库的模块化等。

这本书是否适合你

本书语言通俗幽默，内容循序渐进，适合所有 Java 初学者阅读。即使没有任何编程基础的读者，也可以轻松阅读本书。

本书与作者的另一本书《Java 面向对象编程》一书相比，前者的特色在于化繁为简，以通俗浅显的语言介绍了 Java 语言的基础知识，并且通过经典有趣的实战演练题帮助读者提高实际编程能力。后者则更为详细全面地阐述了 Java 语言的各种技术、性能优化的策略，以及 Java 的核心思想。前者提纲挈领，后者包罗万象，两者各有千秋，相得益彰。

本书致力于轻轻松松地带领读者跨入 Java 世界的大门，体验通过 Java 编程实现各种实用范例的乐趣，而《Java 面向对象编程》则帮助读者全面掌握 Java 的各种技术，并且更深刻理解 Java 的核心思想，进一步提高 Java 编程能力。

本书的所有范例都基于最新的 JDK9 版本。本书每一章都提供了典型有趣的编程实战题。建议读者首先尝试自己独立完成题目。当遇到困难时，再参考书中给出的“编程提示”。这样会更有助于快速提高你的实际编程能力。

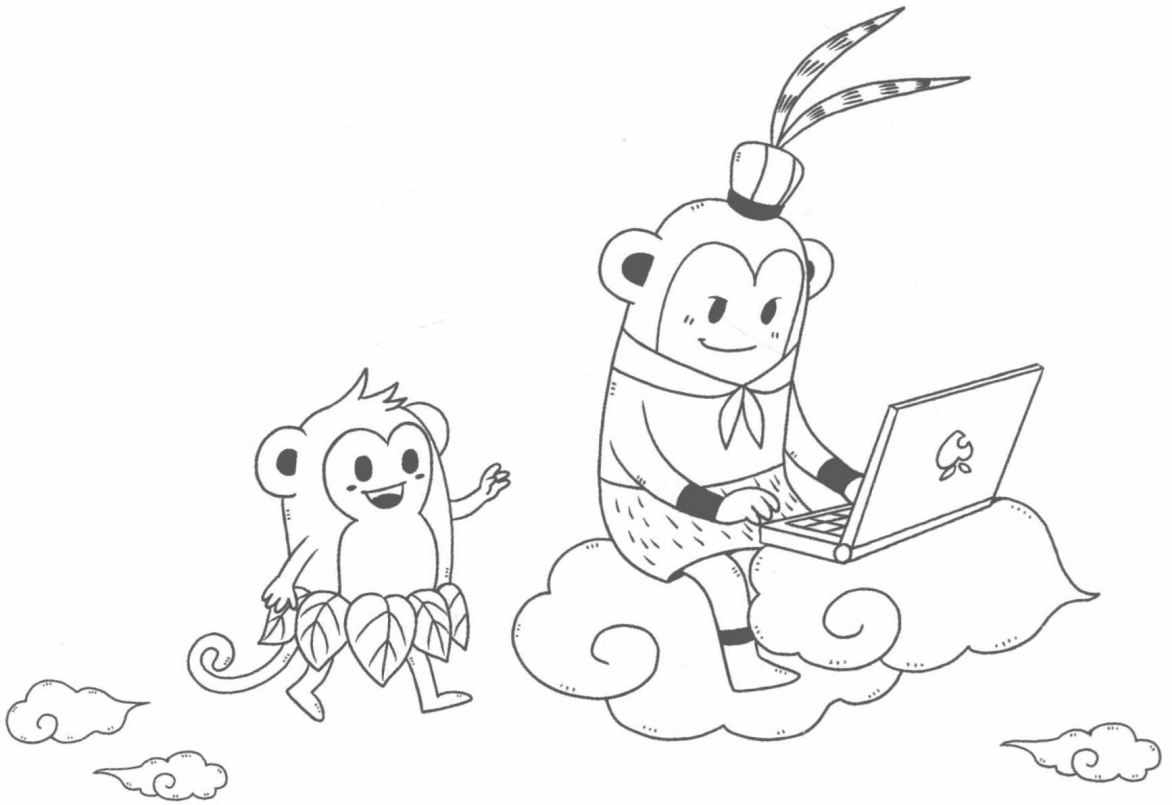
本书技术支持网站

以下是作者为本书提供的技术支持网址，读者可通过它下载与本书相关的资源（包括源代码、软件安装程序、视频教程和讲义等），还可以与作者互动，或者和其他读者交流学习心得，以及对本书提出宝贵意见：

<http://www.javathinker.net/funnyjava.jsp>

致谢

本书在编写过程中得到了 Oracle 公司、电子工业出版社，以及 JavaThinker.net 网站的网友们的大力支持与帮助，在此表示衷心的感谢！参与编写的人员有孙卫琴、张雷、许亮思、张宇客、孟祥、王琨、曹文伟、曹雅洁、李红军、李洪成共十人。尽管我们尽了最大努力，但本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正。



第 1 章 编程入门一点通	1
1.1 初识高级编程语言	1
1.2 跨越平台无障碍, Java 语言 显身手	3
1.3 编写面向对象的 Java 源程序	5
1.3.1 定义 Monkey 类	5
1.3.2 创建 Monkey 对象	6
1.3.3 程序入口 main() 方法	8
1.4 编译和运行 Java 程序	9
1.4.1 JDK 简介	9
1.4.2 本范例的目录结构	11
1.4.3 编译 Java 源程序	11
1.4.4 运行 Java 程序	12
1.4.5 创建用于编译和运行 Java 程序 的批处理文件	13
1.5 用 JDeveloper 软件来开发 Java 应用	13
1.6 小结	15
1.7 编程实战: 八戒用餐怀感恩	16
第 2 章 基本语法了如指掌	19
2.1 Java 源文件结构	19
2.2 关键字	20
2.3 标识符	21
2.4 Java 语言大小写敏感	22
2.5 包声明语句	22
2.6 包引入语句	24
2.7 方法的声明	26
2.8 注释语句	27
2.9 编程规范	28
2.10 JavaDoc 文档	28
2.11 直接用 JDK 来编译和运行 本章范例	29
2.12 用 JDeveloper 来编译和运行 本章范例	31
2.13 用 Eclipse 来编译和运行 本章范例	33
2.14 Java 类库模块化	36
2.15 使用 JShell 交互式编程界面 ..	38

2.16 小结	40
2.17 编程实战: 八戒吃瓜 美滋滋	41
第 3 章 数据类型齐争艳	43
3.1 基本类型	44
3.1.1 boolean 类型	44
3.1.2 byte、short、int 和 long 类型	45
3.1.3 char 类型与字符编码	46
3.1.4 float 和 double 类型	47
3.2 引用类型	47
3.3 基本类型与引用类型的区别 ...	48
3.4 直接数	50
3.4.1 直接数的类型	50
3.4.2 直接数的赋值	51
3.5 小结	51
3.6 编程实战: 金箍棒的 电子档案	52
第 4 章 千姿百态话变量	55
4.1 变量的作用域	55
4.1.1 实例变量和静态变量	57
4.1.2 用静态变量统计实例的 个数	59
4.1.3 成员变量和局部变量同名	60
4.2 对象的默认引用: this	60
4.3 参数传递	61
4.4 变量的初始化及默认值	63
4.4.1 成员变量的初始化	64
4.4.2 局部变量的初始化	65
4.4.3 用 new 关键字创建对象	65
4.5 小结	67
4.6 编程实战: 人参果树妙回春	69
第 5 章 操作符号显身手	71
5.1 操作符的优先级	72
5.2 整型操作符	73
5.2.1 一元整型操作符	73
5.2.2 二元整型操作符	73
5.3 浮点型操作符	74

5.4 比较操作符和逻辑操作符	75
5.4.1 比较操作符	75
5.4.2 逻辑操作符	76
5.5 特殊操作符“?:”	78
5.6 变量的赋值	78
5.7 基本数据类型转换	79
5.7.1 自动类型转换	79
5.7.2 强制类型转换	81
5.8 小结	81
5.9 编程实战: 判断年份是否为 闰年	82
5.10 编程实战: 数字加密	83
第6章 运筹帷幄控流程	85
6.1 分支语句	86
6.1.1 if...else 语句	86
6.1.2 switch 语句	87
6.2 循环语句	91
6.2.1 while 语句	93
6.2.2 do...while 语句	94
6.2.3 for 语句	95
6.3 流程跳转语句	96
6.4 小结	98
6.5 编程实战: 实现常用 数学运算	98
6.6 编程实战: 打印金字塔	99
6.7 编程实战: 考试分数和等级 转换	100
6.8 编程实战: 数兔子	101
6.9 编程实战: 寻找水仙花数	103
第7章 代码重用靠继承	105
7.1 继承的基本语法	107
7.2 方法重载 (Overload)	108
7.3 方法覆盖 (Override)	109
7.4 方法覆盖与方法重载的 异同	111
7.5 super 关键字	112
7.6 多态	113
7.7 小结	115
7.8 编程实战一: 运用方法的重载 和覆盖	115
7.9 编程实战二: 演绎孙悟空与 二郎神斗法	116
第8章 引用类型操作符	121
8.1 字符串连接操作符“+”	121
8.2 操作符“==”与对象的 equals()方法	122
8.2.1 操作符“==”	122
8.2.2 对象的 equals()方法	123
8.3 操作符“!=”	125
8.4 引用变量的赋值和类型 转换	126
8.5 instanceof 操作符	127
8.6 小结	128
8.7 编程实战: 辨别真假 孙悟空	129
第9章 公私分明设权限	131
9.1 封装类的部分属性和方法	132
9.2 4种访问控制级别	133
9.3 小结	136
9.4 编程实战: 模拟自动 洗衣机	137
第10章 abstract: 虚拟抽象画蓝图	143
10.1 abstract 修饰符的 修饰内容	143
10.2 abstract 修饰符的 语法规则	144
10.3 抽象类不能被实例化	145
10.4 小结	146
10.5 编程实战: 金、银角大王 的魔法宝物	146
第11章 final: 一锤定音恒不变	149
11.1 final 类	150
11.2 final 方法	150
11.3 final 变量	151
11.4 小结	153

11.5 编程实战：无法伪造篡改 的生死簿.....	154	15.3 局部内部类.....	198
第 12 章 static：静态家当共分享	157	15.4 匿名类.....	199
12.1 static 变量.....	158	15.5 用 Lambda 表达式代替 内部类.....	201
12.2 static 方法.....	158	15.6 小结.....	202
12.2.1 静态方法可访问的内容.....	159	15.7 编程实战：内部类回调 外部类.....	203
12.2.2 实例方法可访问的内容.....	160	第 16 章 降伏异常有策略	205
12.2.3 静态方法必须被实现.....	161	16.1 Java 异常处理机制概述.....	206
12.2.4 作为程序入口的 main()方法 是静态方法.....	161	16.2 运用 Java 异常处理机制.....	209
12.3 static 代码块.....	162	16.2.1 try...catch 语句： 捕获异常.....	209
12.4 小结.....	163	16.2.2 finally 语句：任何情况下 必须执行的代码.....	210
12.5 编程实战：灵活配置 绘制图形.....	163	16.2.3 throws 子句：声明可能会 出现的异常.....	211
第 13 章 对外开放靠接口	167	16.2.4 throw 语句：抛出异常.....	212
13.1 接口的概念和语法规则.....	169	16.2.5 异常处理语句的 语法规则.....	212
13.2 比较抽象类与接口.....	171	16.2.6 异常流程的运行过程.....	215
13.3 小结.....	173	16.3 Java 异常类.....	216
13.4 编程实战：紧箍圈降伏 诸顽劣.....	174	16.3.1 运行时异常.....	219
第 14 章 出生入死话对象	177	16.3.2 受检查异常 (Checked Exception).....	219
14.1 对象的构造方法.....	178	16.3.3 区分运行时异常和受 检查异常.....	219
14.1.1 重载构造方法.....	179	16.4 用户定义异常.....	221
14.1.2 默认构造方法.....	180	16.5 小结.....	222
14.1.3 子类调用父类的构造方法.....	181	16.6 编程实战：囫途开车 遇异常.....	223
14.2 垃圾回收.....	184	第 17 章 数组元素排排坐	227
14.2.1 垃圾回收的时机.....	185	17.1 数组简介.....	229
14.2.2 对象的 finalize()方法.....	186	17.2 数组变量的声明.....	229
14.3 小结.....	187	17.3 创建数组对象.....	229
14.4 编程实战：玩转垃圾回收.....	188	17.4 访问数组的元素和长度.....	231
14.5 编程实战：独一无二 玉净瓶.....	190	17.5 数组的初始化.....	232
第 15 章 类型封装内部类	191	17.6 数组排序.....	233
15.1 内部类的种类.....	193	17.7 多维数组.....	234
15.2 成员内部类.....	193		
15.2.1 实例内部类.....	194		
15.2.2 静态内部类.....	197		

17.8 用符号“...”声明数目 可变参数.....235	19.3.2 DataInputStream 类.....267
17.9 小结.....236	19.4 输出流.....269
17.10 编程实战：多位数字 加密.....237	19.5 FilterOutputStream （过滤输出流）.....269
17.11 编程实战：用数组实现 堆栈.....238	19.5.1 DataOutputStream.....269
第 18 章 集合元素大操练.....241	19.5.2 BufferedOutputStream.....270
18.1 Java 集合的类框架.....242	19.5.3 PrintStream 类.....270
18.2 集合的基本用法.....242	19.6 Reader/Writer 概述.....271
18.2.1 包装类的自动装箱和 拆箱.....243	19.7 Reader 类.....273
18.2.2 Set（集）和 List（列表） 的各种具体实现类的特点243	19.7.1 InputStreamReader 类.....273
18.2.3 集合的静态 of()方法.....245	19.7.2 FileReader 类.....273
18.3 List（队列）.....247	19.7.3 BufferedReader 类.....274
18.4 Map（映射）.....247	19.8 Writer 类.....274
18.5 用 Lambda 表达式遍历 集合.....249	19.8.1 OutputStreamWriter 类.....275
18.6 小结.....250	19.8.2 FileWriter 类.....275
18.7 编程实战：计算数学 表达式.....250	19.8.3 BufferedWriter 类.....275
18.8 编程实战：计算带括号的 数学表达式.....254	19.8.4 PrintWriter 类.....275
18.9 编程实战：用集合工具对 数字排序.....257	19.9 读写文本文件的范例.....276
18.10 编程实战：按月份先后 顺序数兔子.....258	19.10 随机访问文件类： RandomAccessFile.....278
18.11 编程实战：用映射来存放 学生信息.....259	19.11 File 类.....279
18.12 编程实战：圆桌报数 游戏.....261	19.12 用 java.nio.file 类库来 操纵文件系统.....281
第 19 章 数据出入靠 I/O.....263	19.13 小结.....284
19.1 输入流和输出流概述.....264	19.14 编程实战：替换文本文件 中的字符串.....285
19.2 输入流.....265	19.15 编程实战：批量修改 文件名.....287
19.3 FilterInputStream （过滤输入流）.....266	第 20 章 并发运行多线程.....289
19.3.1 BufferedInputStream 类.....267	20.1 Java 线程的运行机制.....290
	20.2 线程的创建和启动.....291
	20.2.1 扩展 java.lang.Thread 类.....291
	20.2.2 实现 java.lang.Runnable 接口.....293
	20.3 线程的状态转换.....295
	20.3.1 新建状态（New）.....295
	20.3.2 就绪状态（Runnable）.....295
	20.3.3 运行状态（Running）.....295
	20.3.4 阻塞状态（Blocked）.....296

20.3.5 等待状态 (Waiting)	296	22.3.2 BorderLayout 边界布局 管理器	336
20.3.6 死亡状态 (Terminated)	296	22.3.3 GridLayout 网格布局 管理器	338
20.3.7 线程状态转换举例	296	22.3.4 CardLayout 卡片布局 管理器	340
20.4 线程调度	297	22.4 事件处理	342
20.4.1 调整各个线程的优先级	298	22.5 AWT 绘图	345
20.4.2 线程睡眠: Thread.sleep()方法	299	22.6 创建动画	348
20.4.3 线程让步: Thead.yield()方法	300	22.7 菜单	350
20.4.4 等待其他线程结束: join()	301	22.8 小结	352
20.5 获得当前线程对象的引用	302	22.9 编程实战: 创建数学 计算器	354
20.6 小结	303	22.10 编程实战: 创建 BMI 指数 计算器	355
20.7 编程实战: 孙悟空偷吃 蟠桃	304		
第 21 章 同步通信多线程	307	第 23 章 轻松访问数据库	361
21.1 线程的同步	309	23.1 安装和配置 MySQL 数据库	362
21.1.1 同步代码块	312	23.2 JDBC API 简介	364
21.1.2 线程同步的特征	314	23.3 JDBC API 的基本用法	367
21.2 线程的通信	316	23.4 获得新插入记录的主键值	370
21.3 小结	320	23.5 封装连接数据库的细节	371
21.4 编程实战: 悟空保唐僧 打群妖	321	23.6 处理 SQLException	376
21.5 编程实战: 运动员赛跑	324	23.7 设置批量抓取属性	377
21.6 编程实战: 秒针、分针和 时针的通信	326	23.8 可滚动及可更新的结果集	378
第 22 章 图形界面俏容颜	329	23.9 小结	385
22.1 图形用户界面的构建机制	329	23.10 编程实战: 创建客户 管理器	385
22.2 容器类组件	331		
22.3 布局管理器	333		
22.3.1 FlowLayout 流式布局 管理器	335		



第 1 章 编程入门一点通

话说我的本家孙悟空帮助唐僧到西天取到真经后，就在天上逍遥自在地当起了斗战胜佛。斗转星移，岁月如梭，一股信息化浪潮席卷全球，悟空的家乡花果山也与时俱进，处处配备了新式的计算机。

如今，悟空会熟练地运行安装在 Windows 操作系统中的各种可执行程序，利用它们来完成特定任务。例如通过浏览器程序来上网，通过记事本程序来编辑文档，通过画图程序来画画，通过计算器程序来进行数学运算。

有一天，悟空正在网上东游西逛，花果山的小猴智多星跑过来，对悟空说：“孙爷爷，我看这计算机上的程序都是给人玩的，要是您也能编写点程序出来，专门给俺们猴儿耍耍，那该多好啊。”

智多星的想法正合悟空的心意。悟空想：要是自己学会了编程，就可以开发出符合猴子趣味的程序给儿孙们耍耍，等到编程功底扎实了，还可以给花果山也开发个网站呢。

在本章中，悟空小试牛刀，用 Java 语言编写了一个简单的程序。本章内容主要围绕以下问题展开：

- 为什么 Java 语言具有跨操作系统平台的特性？
- 什么是面向对象（Object Oriented，OO）的基本思想？
- 创建、编译和运行 Java 程序的基本过程是怎样的？

1.1 初识高级编程语言

当两个人使用不同的语言时，如果要相互沟通，就必须请翻译员充当沟通的桥梁，有时甚至要经过好几个翻译员的翻译。例如，佛祖最初在印度讲佛经采用的是梵语，后来由唐僧翻译成中文，再后来鉴真和尚又把中文的佛经翻译成日文，传到日本。

首先，悟空想让计算机模仿智多星说话。如图 1-1 所示，悟空先用猴语幽默地对计算机大声喊：“嘿，伙计，帮我在屏幕上打印一行字符串‘大家好，我是智多星’。”可是计算机置若罔闻，它又不是猴脑，哪懂得悟空的猴语啊。悟空宽宏大量地对计算机笑笑，经过西天取经的磨炼，悟空已经改了动不动就亮出金箍棒吓唬人的毛病。他明白，要与对方交流，首先要熟悉对方使用的语言，实在学不会，就得请个翻译员。



图 1-1 悟空试图用猴语与计算机对话

计算机作为硬件，只懂得由“1”和“0”排列组合成的机器指令语言。机器指令语言又复杂又枯燥，悟空可不想学呆板的机器指令语言。幸运的是，悟空可以不必直接用机器指令语言和计算机对话，而是请计算机中的“翻译员”——操作系统来进行沟通。

下面以悟空访问 IE 浏览器程序为例，介绍操作系统如何在悟空和计算机之间充当“翻译员”。如图 1-2 演示了悟空运行 Windows 操作系统中的 IE 浏览器程序的过程。悟空只需双击 IE 浏览器程序图标，聪明的 Windows 操作系统就明白了悟空的意图，它就会与计算机交互，请求计算机运行 IE 浏览器程序。Windows 操作系统就像悟空与计算机之间的翻译员。

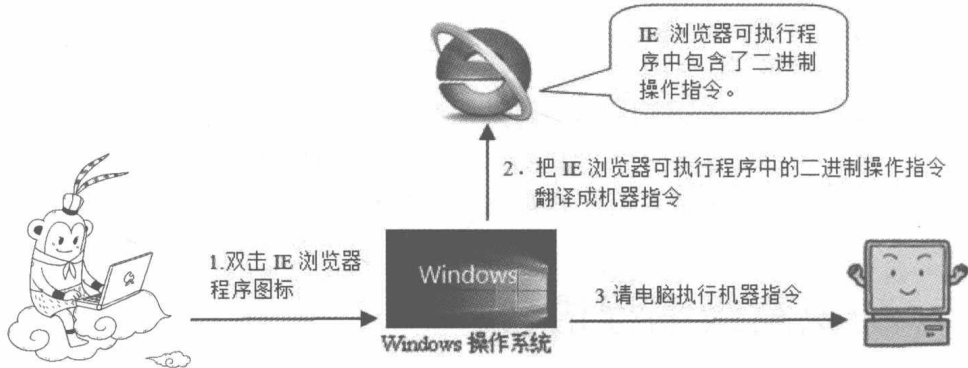


图 1-2 悟空与 Windows 操作系统对话，让其运行 IE 浏览器程序

从图 1-2 可以看出，大总管 Windows 操作系统八面玲珑，精通多种语言，既能与悟空沟通，还能读懂 IE 浏览器程序中的二进制操作指令，并能把二进制操作指令翻译成机器指令，最后用这种机器指令语言对计算机发号施令。

悟空如果想让计算机模仿智多星说话，只需要编写一个模拟智多星说话的程序，接下来让操作系统来运行这个程序就行了。以 Windows 操作系统为例，它的可以执行的程序（简称可执行程序）通常都是以“.exe”作为扩展名的文件。这些可执行程序和 IE 浏览器程序一样，包含了二进制的操作指令，这些操作指令只有 Windows 操作系统才能看得懂。

如图 1-3 所示，模拟智多星的可执行程序中包含的是二进制的操作指令，可是悟空根本没有耐心去学习这些和机器指令语言一样枯燥乏味的操作指令。

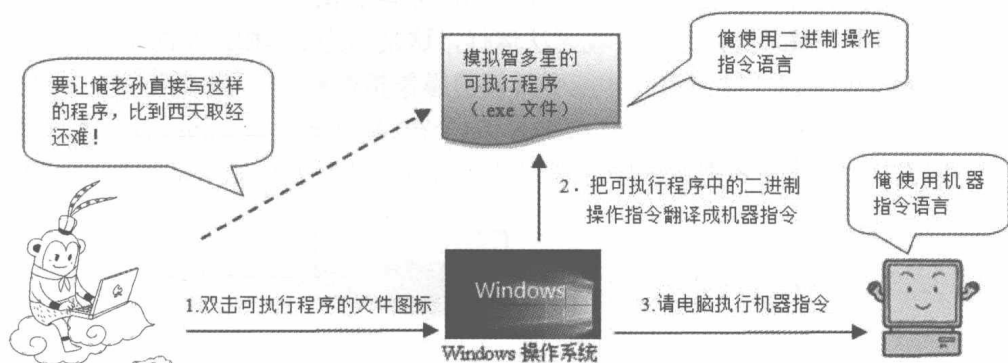


图 1-3 悟空与 Windows 操作系统对话，让其运行模拟智多星的可执行程序

幸运的是，悟空可以用高级编程语言来编程，高级编程语言与人类语言在语法上更加接近，比较容易掌握。可是，操作系统并不懂高级编程语言，因此还必须想办法把用高级编程语言编写出来的源程序转换为操作系统看得懂的可执行程序，这个过程叫作编译。

如图 1-4 所示，悟空用高级编程语言编写了一个模拟智多星的源程序，接下来用现成的编译器软件把源程序编译为可执行程序，然后让操作系统来运行这个可执行程序。

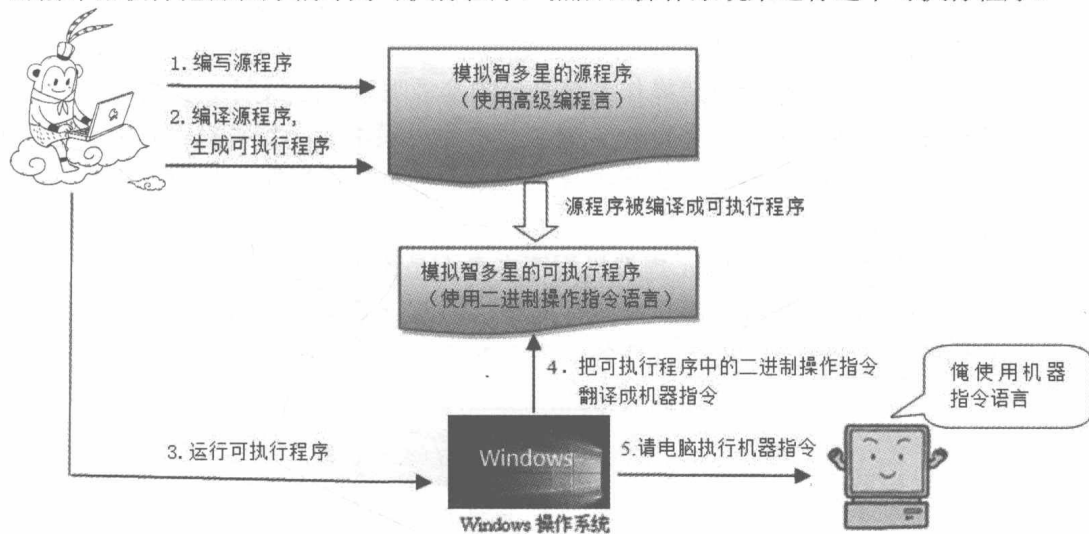


图 1-4 程序的编写、编译和运行过程

从图 1-4 可以看出，悟空要想让计算机能听从它的各种吩咐，主要的任务就是掌握一门高级编程语言，然后用它来编写源程序。

1.2 跨越平台无障碍，Java 语言显身手

在本章 1.1 节讲过，操作系统看不懂用高级编程语言编写出来的源程序，但是能看得懂编译生成的可执行程序。那么，对于同一个可执行程序文件，是不是所有的操

作系统都能看得懂呢？答案是否定的。例如，IE 浏览器只能在 Windows 操作系统中运行，到了 Linux 操作系统中就无法运行，这是因为 IE 浏览器的可执行程序文件中包含了只有 Windows 操作系统才能看得懂的操作指令。

如图 1-5 所示，假如悟空想编写一个在 Windows 和 Linux 操作系统中都能运行的程序，那么就需要把源程序分别编译成适合这两种操作系统的可执行程序。

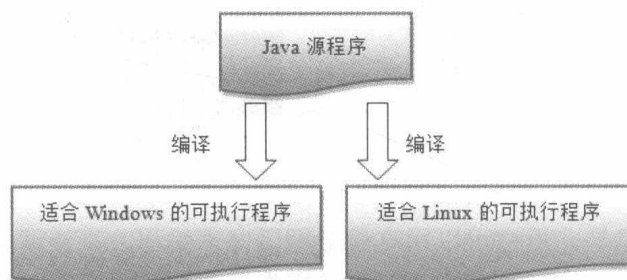


图 1-5 把源程序分别编译成适合 Windows 和 Linux 的可执行程序

悟空觉得这样做还是有点麻烦，要是有一种编程语言可以跨操作系统平台就好了，这意味着只需对用这种语言编写出来的源程序编译一次，编译出来的可执行程序能够在所有的操作系统中运行。刚好 Java 语言就是悟空所期望的跨操作系统平台的高级编程语言。

Java 语言为何会有跨操作系统平台的本领呢？这还得归功于 Java 虚拟机。Java 虚拟机这名字听上去很玄乎！Java 虚拟机看不见摸不着，到底算何方神圣？它可不是工厂里庞大无比的机器，其实它本身也不过是个可执行程序，这个可执行程序的任务就是运行 Java 程序。

如图 1-6 所示，Java 虚拟机程序本身不是跨操作系统平台的，对于不同的操作系统，有着不同的 Java 虚拟机可执行程序。不过，不管是哪个操作系统中的 Java 虚拟机，它们的任务都是一样的，该任务就是请求底层操作系统运行 Java 程序。

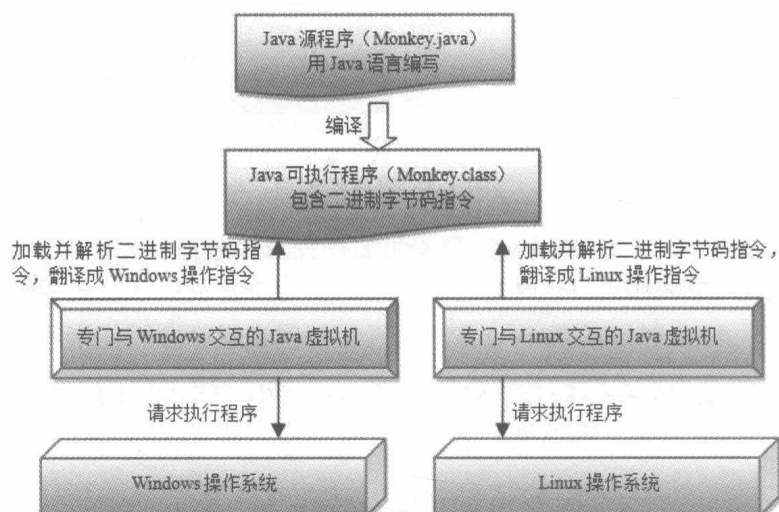


图 1-6 Java 程序的跨操作系统平台运行的过程

从图 1-6 可以看出, Java 源程序是以“.java”作为扩展名的文件, 编译生成的可执行程序是以“.class”作为扩展名的文件。Java 可执行程序本身不能直接在操作系统中运行, 它必须借助 Java 虚拟机才能运行。Java 可执行程序中包含只有 Java 虚拟机才能看得懂的二进制字节码指令, 而 Windows 和 Linux 操作系统都无法直接看得懂这些二进制字节码指令。

1.3 编写面向对象的 Java 源程序

悟空经过一番认真学习, 掌握了 Java 语言的基本语法, 接着就猴急猴急地要在儿孙们面前演示 Java 语言的神奇妙用了。有一天, 悟空把智多星叫过来, 故作神秘地对智多星说: “请你如实回答我的以下问题。”

悟空: “你属于猫类、鸡类、狗类, 还是猴类啊?”

智多星: “当然是猴类啦。”

悟空: “你叫什么名字?”

智多星: “智多星呀。”

悟空: “你能做哪些事?”

智多星: “吃饭、说话、睡觉、爬树、杂耍……”

悟空接下来得意地对智多星说: “我要把你搬到计算机里。”智多星惊讶地瞪大了眼睛: “孙爷爷啊, 您神通广大, 能把金箍棒变成针眼般大小, 再把它塞到耳朵里。莫非您也要把孩子变成芝麻般大小, 再把孩子塞到这计算机里?”悟空哈哈大笑: “你想歪啦。我要在计算机里创建一个虚拟的智多星, 它也能像你一样说话。”智多星兴奋得跳了起来: “太好了, 孩儿以前在计算机里看电视剧《西游记》, 看到有您、唐僧和猪八戒等, 还有白骨精呢, 唯独没有我。这回我终于也能在计算机里露个脸了。”

1.3.1 定义 Monkey 类

悟空打开文本编辑器 (例如 Windows 中的记事本程序), 用 Java 语言编写了一个如例程 1-1 所示的 Java 源程序, 文件名为 Monkey.java。

例程 1-1 Monkey.java

```
public class Monkey{ //定义代表花果山猴类的 Monkey 类
    String name; //定义名字属性

    public Monkey(){ //不带参数的构造方法

    public Monkey(String name){ //带参数的构造方法
        this.name=name; //设置 Monkey 对象的 name 属性
    }

    public void speak(){ //定义模拟猴子说话的 speak()方法
        //猴子给大家打招呼
        System.out.println("大家好, 我是"+name);
```

```
}
}
```

在这个 Monkey.java 文件中定义了一个 Monkey 类，它代表花果山上的猴类。Monkey.java 文件主要由以下内容构成：

(1) 类的声明语句：

```
public class Monkey{...}
```

以上代码指明类的名字为“Monkey”，public 修饰符意味着这个类可以被公开访问。类的主体内容都位于大括号“{}”内。



提示：

在本书中，“声明”和“定义”有着相同的含义。例如，“声明 Monkey 类”和“定义 Monkey 类”的意思相同；“声明 name 变量”和“定义 name 变量”的意思也相同。

(2) 类的属性（也称为成员变量）的声明语句：

```
String name;
```

所有的猴子都有名字，用 Monkey 类的 name 属性来表示。name 属性为 String 字符串类型。

(3) 方法的声明语句和方法主体：

```
public void speak(){
    System.out.println("大家好，我是"+name);
}
```

所有的猴子都具有说话的行为，Monkey 类的 speak() 方法就用来模拟猴子说话的行为。speak() 方法的大括号中的内容称为方法主体，以上方法主体中程序代码的作用是打印“大家好，我是 XXX”。



提示：

Java 语言是区分大小写的。例如“Monkey”和“monkey”是两个不同的名字。如果读者要重新编写 monkey 类的源代码，要注意不要混淆代码中字符的大小写。

Monkey.java 文件中以“//”开头的文字代表注释，它不是程序代码，而是用于解释说明程序代码，从而便于编程人员理解程序代码。对于 Monkey.java 文件中 public 和 static 等修饰符，暂且不用细究它们的用途。到目前为止，读者只要大致了解 Monkey 类主要由 name 属性和 speak() 方法构成就行了。

1.3.2 创建 Monkey 对象

在上一节，悟空已经定义了一个 Monkey 类。这个 Monkey 类是所有猴子的模板，每个猴子都是 Monkey 类的一个实例，例如猴子智多星就是 Monkey 类的一个具体实例。Monkey 类具有 name 属性和 speak() 方法，那么依照 Monkey 类模板生成的所有 Monkey 实例也会具有 name 属性和 speak() 方法。例如猴子智多星的 name 属性的值为“智多星”，speak() 方法的作用是打印字符串“大家好，我是智多星”。