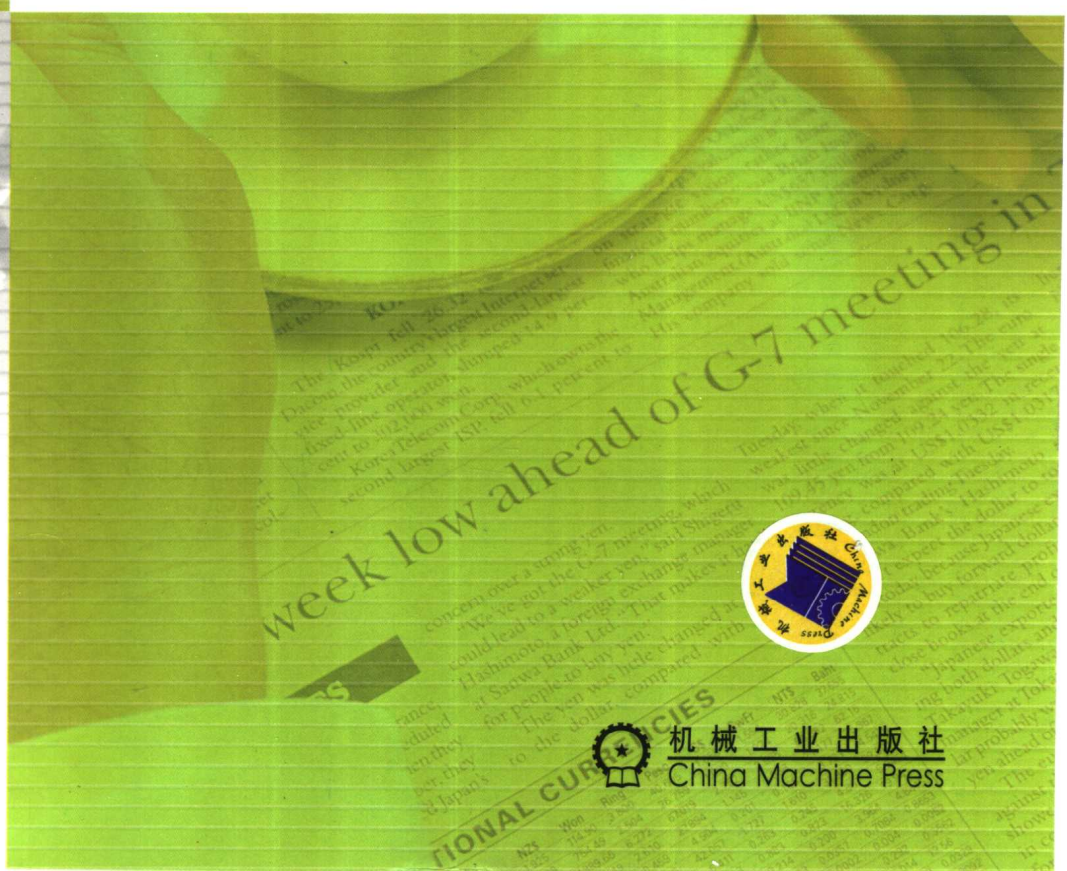
内附光盘

Java 阶梯丛书

Java 基础教程

■ 樊荣 编著



机械工业出版社
China Machine Press

Java 阶梯丛书

Java 基础教程

樊 荣 编著

机 械 工 业 出 版 社

本书主要讲解 Java 语言的核心基础知识，使读者能有个良好的语言基础，为以后进一步提高打下基础。

作者在长期的 Java 语言教学中，总结出初学者、一般开发人员不容易理解的概念、知识要点，通过大量的有针对性的简单例子和丰富通俗的讲解，帮助读者清晰全面地了解 Java 语言的最核心概念和知识。

本书的理念是从实践中学习，从代码中理解概念，让读者从程序实践和感性认识上升到对核心概念的理解，重点讲解 Java 语言核心知识。书中的内容涵盖了 Java 语言基础、面向对象编程、异常处理、图形化用户界面、I/O、线程以及网络编程等。

本书的主要对象是 Java 语言的初学者以及希望能够比较全面地理解 Java 语言的读者。

图书在版编目（CIP）数据

Java 基础教程 / 樊荣编著.

-北京：机械工业出版社，2004.5

（Java 阶梯丛书）

ISBN 7-111-14341-8

Ⅰ.J… Ⅱ. 樊… Ⅲ. Java 语言-程序设计

Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2004）第 032825

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：吴宏伟　　版式设计：谭奕丽

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2004 年 5 月第 1 版第 1 次印刷

787mm×1092mm 1/16 · 21.75 印张 · 520 千字

0001-5000 册

定价：34.00 元（含 1CD）

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

前　言

Java 学习

读者选择这本书是因为对 Java 感兴趣，希望选择 Java 语言作为自己职业生涯的发展方向。怎样的选择决定怎样的结果，选择很重要，所以有许多人在做选择上花了很多时间，不能确定自己该做什么，因为软件开发的面非常广，相关技术比较多，而自己对这些技术不了解，无从选择，有些人甚至因为不知道选什么就放弃了学习。果断地作出选择，Java 技术是进入 IT 行业的一个很好的切入点。

坚定一下大家的信念：进入 IT 行业，从 Java 开始是正确的选择

首先，是金钱和人力。IT 巨头如 IBM、SUN、ORACLE 及 BEA 等大公司投入了大量的人力和物力推动这个技术的发展，没有希望的东西，是不会有人投资的。

其次，Java 是一个免费的技术，盗版不可能成为永远的免费午餐，而用 Java 可以降低开发成本。

再次，Java 语言的功能强大，应用面十分广泛，从大型的企业应用到手机游戏的开发，都包括在内，而且相对来说比较容易学，它与底层系统的关联不大，开发的程序稳定性高，对初学者来说是一个很好的切入点，通过 Java 语言可以逐步接触到网络、数据库等各种技术。

所有技术甚至所有事物的原理都是相通的，所谓“一法通，百法通”。能够精通 Java，掌握其思想，再去理解其他语言和技术就非常容易了，反之亦然。所以有时需要的不是多想，而是多做、多执行。

关于开发，有些想法不是很正确

一是要学功能最强大的语言，这样不是很好。最强大的语言，应该说是 C 语言，因为它产生有近 40 年的历史，经过了历史的考验，现在仍然充满活力，UNIX 操作系统就是采用 C 语言开发的。但最强的东西不一定是最合适的，C 语言同系统结合紧密，要做开发，不仅要熟悉语言本身同时也需要对相应的操作系统非常熟悉，学习周期比较长，而且因为其功能强大，破坏能力也强大，掌握不熟练，不容易做出很好的商业化程序。C 语言适合做系统开发，如操作系统，但操作系统或其他底层应用类似于硬件。而对应于应用，可能每个公司或者每个人的需求都不一样，它需要的人员更多，市场更大，而且对于 Java 应用来说，功能足够强大。

二是感觉做开发不长久，技术更新快，太辛苦了，而且做到 30 岁左右就会没有人要了。虽说科技的发展是一日千里，但真正的创新，真正的创造性的东西并不是很多。如航空母

舰，半个世纪了，并没有巨大的变化。技术的更新，大部分时候也只是一些小的改进而已，而且万变不离其宗，如果把技术掌握得够好，新的技术是非常容易消化的。至于说做开发年纪大了没有人要，这更加错误了，因为国内的软件开发起步晚，所以目前没有多少年纪大的人做开发。技术经验是积累性的，从业时间越长，年纪越大，其能力也会越强，到时候不是没有人要，而是很多人抢。现在国内就是缺乏真正资历深厚的开发人员。

三是过于关注某个技术点，而不重视基础。大部分人要学 Java 只是知道它热门，大概知道 J2EE、J2ME 就来学。这些技术也不难，做一些简单的开发很容易，但要做完整的商业应用时，或者出了一点问题时，就不知道该怎么办了。只知其表，不知其本质和思想，不少做 Web 开发两三年的人，Servlet、JSP 都会用，但连基本的 HTTP 协议都不知道，这样又怎么能流畅地开发出应用来呢？现在很多技术开发出来的目的就是为了开发快、简单，所以封装再封装，其底层的东西、原理都不知道，这样不便于灵活运用，技术再先进、再灵活、再智能，也不能与人相比。只有从基础、本质和思想理解了某个技术才能真正灵活运用。这里，语言的基础是最重要的，有了良好的基础，才能天高任鸟飞，海阔凭鱼跃。

怎样学 Java

像做任何事一样，首先要有恒心和热情。如果日思夜想、持之以恒，学 Java 就容易了。道理很简单，做到却不容易。

一个非常重要的方法就是多练习，多写代码，从错误中学习，从实验中学习。同很多其他科学一样，做实验是最重要的，实验做多了，现象看多了，自然能上升到理论上来，从现象来理解概念就非常容易了。如果只是看看书，听听课，像大学上课一样，不动手，是很难掌握一门语言的。许多概念不清楚，自己尝试写一个程序，看它怎么运行，运行结果是什么，就很容易理解了。学 Java 编程，要把自己当成是一个实验员，不断地写代码，做实验，才能快速提高自己。程序员是写代码写出来的，系统设计分析员也是写代码写出来的。

了解了基本的概念后，开始做具体的项目，在项目中实践应用才能进一步提高自己。在刚开始做项目时，最好不要自己去做，要坚信一点，所做的东西 99%是有人做过的，更资深的人做过，在网上是一定能找到的，要去找这些资源，看人家的代码，理解人家的代码，再修改为自己所用，站在人家的肩膀上前进，比自己一步一步地爬要快得多。Internet 是个巨大的资源库，而 Google 就是找所需要的资源的捷径。等要再进一步地提高自己时，可以去看很多公开源代码的项目，APACHE.ORG 提供了非常多的东西让我们学习借鉴。

在学技术之前，要先学会语言本身，不要只是去看 SWING、EJB、Servlet 及 JMS 等技术，把语言基础打好了，甚至能自己写一个 TOMCAT、一个 Web Server，再去了解这些技术就非常容易了。如果只是能用这些技术，而不知道到底发生了什么，则很难做到深入灵活地运用。

在应用技术的过程中，也需要理解其本质思想，而不能只见树木不见森林。如对象、数据库表、XML、协议及类的定义、XML 的 DTD、协议的定义，都可以理解为数据库表的定义、对象、XML 记录，一个协议数据包就是表的具体记录。但它们具体的应用又有所

不同，再去理解就比较容易了。在面向对象的语言里，重要的是把面向对象的思想领会，可以在了解了语言基础后，接触一下设计模式，来体会面向对象的思想。

做 Java 开发是一定要用到外语的，看书看资料，最新的东西或者说稍微深入一点的东西都是英文的。java doc 的开发文档也是英文的，里面的内容是最有用、最权威的。如果基本概念掌握得很好，英语又熟练，基本不需要其他资料了，只要多看看 Java API .doc 即可。否则，一门技术出来，相关资料要等人家翻译。又能翻译，又懂技术的人不多。而且中文的 Java 资料中，作者真正自己把内容彻底消化，然后再写出来不是很容易，要做到全面、系统就更不容易了。所以学 Java 要常看英文资料，特别是 Java 的 doc。其实 IT 英语主要是一些专业词汇，没有什么太多语法，是比较容易学的，或者有个金山词霸，一边看一边查，养成好的习惯就好了。

关于本书

基于基础就是核心，全面清晰深入地理解核心概念和思想这一指导思想，本书专门讲 Java 语言本身，而不是讲具体技术的书。侧重于讲解重要的 Java 核心概念和基础知识，把一般开发人员不容易理解的概念作比较全面和深入的讲解，而那些概念不多的内容，查查文档就能明白的内容，本书就不细述。

本书通过大量的例子来帮助大家理解概念，围绕概念、例子和理解 3 个方面展开，适合初学者或基础知识不是很牢固的 Java 爱好者使用。

关于作者

樊荣：从事 Java 相关开发并任培训机构的 Java 教师多年，曾为许多企业做 Java 相关应用的培训和咨询。由于时间仓促及本人的水平有限，书中难免有错误之处，欢迎大家批评指正，电子邮件地址：bill_fan@21cn.com。

目　录

目 录

第 1 章　Java 简介与开发环境设置

本章介绍 Java 的历史现状和特点，以及 Java 运行环境的安装和设置。如果读者知道基本的安装设置，可以直接跳到第 2 章开始学习，通过具体的程序了解了 Java 后，有了感性认识，再回到这一章，看 Java 的特性介绍，就比较好理解了。

1.1　Java 产生的历史

Java 来自于 Sun 公司的一个称为 Green 的项目，其最初的目的是为家用消费电子产品开发一个分布式代码系统，这样用户可以把 E-mail 发给电冰箱、电视机等家用电器，对它们进行控制，和它们进行信息交流。开始，准备采用 C++语言，但 C++太复杂，安全性差，最后基于 C++开发一种新的语言 Oak（Java 的前身），Oak 是一种用于网络的精巧而安全的语言。

Java 的取名也有趣闻，一天，几位 Java 成员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java（爪哇）咖啡，有一个人灵机一动说就叫 Java 怎样，得到了其他人的赞赏，于是，Java 这个名字就这样传开了。

1.2　Java 的现状

Java 是 Sun 公司推出的新的一代面向对象程序设计语言，特别适合于网络通信应用程序开发，它的最大特点是平台无关性。

随着 Java 编程的应用变得越来越广泛，Java 语言目前成为技术开发中最常用的一种语言。Java 作为软件开发的一种革命性技术的地位已被确立，这表现在以下几个方面：

（1）计算机产业的许多大公司购买了 Java 的许可证，包括 IBM 、Apple、DEC、Adobe、SiliconGraphics、HP、Oracel、Toshiba 以及 Microsoft。

（2）众多的软件开发商大力支持 Java 的软件产品。例如：Borland 公司开发了基于 Java 的快速应用程序开发环境 Jbuilder，目前最新的版本为 Jbuilder 9.0。

（3）Intranet 正在成为企业信息系统最佳的解决方案，而其中 Java 将发挥不可替代的作用。Intranet 的目的是把 Internet 用于企业内部的信息系统，它的优点表现在便宜、易于使用和管理。用户不管使用何种类型的机器和操作系统，界面是统一的 Internet 浏览器，而数据库、Web 页面、应用程序（用 Java 编的 Applet）则存在 WWW 服务器上，无论是开发人员、管理人员，或是用户都可以受益于该解决方案。

Java 语言正在不断发展和完善，Sun 公司是主要的发展推动者，较通用的编译环境有

JDK（JavaDevelopKit）。还有很多其他公司正在开发 Java 语言的编译器与集成环境，预计不久 Java 语言的正确性与效率都将会提高，用户用 Java 编程和现在用 C++编程一样方便。

1.3 Java 语言的特点

Java 到底是一种什么样的语言呢？Java 是一种简单的、面向对象的、分布式的、解释的、健壮的、安全的、结构中立的、可移植性能很优异的、多线程的、动态的语言。

1．简单

Java 最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。Java 语言的简单性主要体现在以下 3 个方面：

（1）Java 的风格类似于 C++，因而是 C++程序员非常易于熟悉的。从某种意义上讲，Java 语言是 C 及 C++语言的一个变种，因此，C++程序员可以很快就掌握 Java 编程技术。

（2）Java 摒弃了 C++中容易引发程序错误的地方，如指针和内存管理。

（3）同时提供了丰富的类库和第三方开发包，以及大量的基于 Java 的公开源代码的项目，可以供我们参考学习。

2．面向对象

面向对象是 Java 最重要的特性。Java 语言的设计完全是面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码继承及重用。单从面向对象的特性来看，Java 类似于 SmallTalk，但其他特性，尤其是适用于分布式计算环境的特性远远超越了 SmallTalk。

3．分布式

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库，以及 RMI 远程调用、EJB 分布式对象调用管理等。因此，Java 应用程序可打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。特别是为分布环境，尤其是 Internet，提供动态内容无疑是一项非常宏伟的任务，但 Java 的语法特性却使这项目标比较容易实现。

4．健壮

Java 能够检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。Java 自己操纵内存，减少了内存出错的可能性。Java 还实现了真数组，避免了覆盖数据的可能。这个功能特征大大缩短了开发 Java 应用程序的周期。

5．结构中立

为了让 Java 作为网路的一个整体，Java 将它的程序编译成一种结构中立的中间文件格式，只要有 Java 运行系统的机器都能执行这种中间代码。Java 源程序被编译成一种高层次的与机器无关的 byte-code 格式语言，这种语言被设计在虚拟机上运行，由机器相关的运行

调试器实现执行。

6．安全

Java 的安全性可从两个方面得到保证：一方面，在 Java 语言里，像指针和释放内存等 C++功能被删除，避免了非法内存操作；另一方面， Java 语言在执行前要经过很多次的测试。它经过代码校验，检查代码段的格式，检测指针操作，对象操作是否过分以及试图改变一个对象的类型。

如果 byte-code 通过代码校验，没有返回错误，则代码没有堆栈上溢出和下溢出、所有操作代码参数类型都是正确的、没有发生非法数据转换（如将整数转换成指针），访问对象操作是合法的。

ClassLoader 通过将本机类与网络资源类的名称分开来保持安全性。因为调入类时总要经过检查，这样避免了特洛伊木马现象的出现。从网络上下载的类被调进一个与源相关的私有的名字域。当一个私有类访问另一个类时，build-in（本机类）首先被检查，然后检查相关的类。这样就避免了破坏本机类情况的出现。

7．可移植的

同体系结构无关的特性使得 Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行，这成为 Java 应用软件便于移植的良好基础。但仅仅如此还不够。如果基本数据类型设计依赖于具体实现，也将为程序的移植带来很大不便。例如，在 Windows 3.1 中整数（Integer）为 16bits，在 Windows 95 中整数为 32bits，在 DECAlpha 中整数为 64bits，在 Intel486 中整数为 32bits。通过定义独立于平台的基本数据类型及其运算，Java 数据得以在任何硬件平台上保持一致。Java 语言的基本数据类型及其表示方式如下：

- byte8-bit　二进制补码
- short16-bit　二进制补码
- int32-bit　二进制补码
- long64-bit　二进制补码
- float32-bitIEEE754　浮点数
- double32-bitIEEE754　浮点数
- char16-bitUnicode　字符

在任何 Java 解释器中，数据类型都是依据以上标准具体实现的。因为几乎目前使用的所有 CPU 都能支持以上数据类型、8～64bits 整数格式的补码运算和单/双精度浮点运算。Java 编译器本身就是用 Java 语言编写的。Java 运算系统的编制依据 POSIX 方便移植的限制，用 ANSIC 语言写成。Java 语言规范中也没有任何同具体实现相关的内容。

8．解释的

Java 解释器（运行系统）能直接运行目标代码指令。链接程序通常比编译程序所需资源少，所以程序员可以在创建源程序上花上更多的时间。

9．高性能

如果解释器速度不慢，Java 可以在运行时直接将目标代码翻译成机器指令。Sun 用直接解释器 1 秒钟内可调用 300000 个过程。翻译目标代码的速度与 C/C++的性能没什么区别。

10．多线程

Java 提供的多线程功能使得在一个程序里可同时执行多个小任务。线程——有时也称小进程，是一个大进程里分出来的小的独立的进程。因为 Java 实现了多线程技术，所以比 C 和 C++更键壮。多线程带来的更大的好处是更好的交互性能和实时控制性能。当然实时控制性能还取决于系统本身（UNIX、Windows、Macintosh 等），在开发难易程度和性能上都比单线程要好。

11．动态

Java 的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行过程中所需要的类，这是 C++语言进行面向对象程序设计所无法实现的。在 C++程序设计过程中，每当在类中增加一个实例变量或一种成员函数后，引用该类的所有子类都必须重新编译，否则将导致程序崩溃。Java 从如下几方面采取措施来解决这个问题。

（1）Java 编译器不是将对实例变量和成员函数的引用编译为数值引用，而是将符号引用信息在字节码中保存下来传递给解释器，再由解释器在完成动态连接类后，将符号引用信息转换为数值偏移量。这样，一个在存储器生成的对象不在编译过程中决定，而是延迟到运行时由解释器确定。这样，对类中的变量和方法进行更新时就不至于影响现存的代码。解释执行字节码时，这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一次，随后代码便可以全速执行。在运行时确定引用的好处是可以使用已被更新的类，而不必担心会影响原有的代码。如果程序连接了网络中另一系统中的某一类，该类的所有者也可以自由地对该类进行更新，而不会使任何引用该类的程序崩溃。

（2）Java 还简化了使用一个升级的或全新的协议的方法。如果系统在运行 Java 程序时遇到了不知怎样处理的程序，Java 能自动下载所需要的功能程序。

1.4 与 C 和 C++语言的异同

Java 提供了一个功能强大的语言的所有功能，且几乎没有一点含混特征。C++安全性不好，但 C 和 C++还是被大家所接受，所以 Java 设计成 C++形式，让大家很容易学习。Java 去除了 C++语言的许多功能，让 Java 的语言功能很精炼，并增加了些很有用的功能。Java 去除了以下几个 C 和 C++功能和特征：指针运算、结构 typedefs#define、需要释放内存全局变量的定义，这些功能都是很容易引起错误的地方。

1.5　Java 的应用简介

Java 是一种与平台无关的语言，因此用 Java 开发的网络应用系统可以在各种平台上运行，大大增加了开发效率，减少重复劳动。而且，Java 集成的网络功能有利于充分开发网络应用系统。

J2SE 最早的名字叫 JDK，最早的 1.0 版约在 1996 年 2、3 月开发出来，1.2 是在 1998 年 12 月份开发出来的。在出 1.2 版本时，就出现了 J2EE 和 J2ME，于是开始出现 2 大块，统称为 Java 2。

- Java 2 Enterprise Edition（J2EE）　定位于服务器端程序的应用。
- Java 2 Standard Edition（J2SE）　定位于客户端程序的应用。
- Java 2 Micro Edition（J2ME）　定位于嵌入式系统的应用。

在 3 种平台中，J2EE 被认为是成长最快的技术应用框架，全球 87%的新应用都是由 J2EE 分布式应用体系支撑的。作为开发多层企业应用的标准，J2EE 充分基于标准化和模块化的组件、提供全面的组件服务并自动处理许多应用行为，从而达到简化企业应用的目的。

J2SE 在客户端提供的许多应用支持特性则包括支持“一次写入，任意运行”的可移植性、面向数据库访问应用的 JDBC API、支持与现有企业资源交互的 CORBA 技术和保护网上数据的安全机制。

J2ME 的策略是为手机赋予本地计算能力。例如，当用户出差在外时，他们可以利用任何可与 Internet 连接的计算机来安全地访问他们的个人信息和应用。与其他嵌入式支持模式不同的是，J2ME 将运算程序直接下载到用户的移动设备上，从而减少用户的数据交互环节，加快应用的便捷性。

Sun 公司计划每隔 18 个月左右就利用新的 API 和功能构建重要版本（名为特性版本）。J2SE 1.3 于 2000 年 5 月交付。2002 年，J2SE 1.4 完成最终版本。代号为 Tiger 的 1.5 版本在 2003 年推出，本书基于 J2SE 的 1.4 版本。

1.6　安装设置 Java 编译运行环境

（1）下载 JDK，访问 Sun 公司的网站: http://java.sun.com/j2se/，如图 1-1 所示。

（2）选择 J2SE Downloads，如图 1-2 所示。

（3）选择 Windows 版 JDK 下载，然后双击“安装”图标，如图 1-3 所示。

（4）安装完后，在控制面版中设置环境变量，如图 1-4 所示。

（5）将 Java 路径下的 BIN 完整路径加到 Path 环境变量即可，这里不需要设置 ClassPath 环境变量。这样就可以用 Java 编程了。