

Java Language & Java Platform

Java 语言与 Java 平台

Pocket Reference



O'REILLY®

中国书籍出版社

David Flanagan 著
洪峰 译

Java 语言与 Java 平台

David

TP312JA/1c

出版社

O'REILLY

00025589

TP312.3A

75

Java 语言与 Java 平台

David Flanagan 著

洪峰 译

O'REILLY®

Beijing • Cambridge • Köln • Paris • Sebastopol • Tokyo

中国书籍出版社

图书在版编目 (CIP) 数据

Java 语言与 Java 平台 / (美) 弗兰尼根 (Flanagan, D.) 著; 洪峰译. - 北京: 中国书籍出版社, 1999. 6

书名原文: Java Language & Java Platform

ISBN 7-5068-0559-6

I. J ... II. ①弗 ... ②洪 ... III. Java 语言 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 13075 号

北京市版权局著作权合同登记

图字: 01-1999-0810 号

Simplified Chinese Edition © China Book Press 1999. Authorized translation of the English edition, 1999 O'Reilly and Associate, Inc., This translation is published and sold by permission of O'Reilly and Associate, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights reproduction in whole or in part in any form.

简体中文版 中国书籍出版社 1999. 授权英文译文, 1999, 奥莱理有限公司。此译本的出版和销售得到出版者和销售权的所有者——奥莱理有限公司的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / Java 语言与 Java 平台

书 号 / ISBN 7-5068-0559-6/TP · 21

责任编辑 / 蔡惠斌 张立

责任印制 / 王大军 刘颖丽

封面设计 / 张健

出版发行 / 中国书籍出版社

地 址 / 北京市丰台区太平桥西里 38 号 (邮政编码 100073)

电 话 / (010) 63455164 (总编室) (010) 63454858 (发行部)

经 销 / 全国新华书店

印 刷 / 北京京海印刷厂

开 本 / 787 毫米 × 1092 毫米 32 开本 2.5 印张 49.5 千字

版 次 / 1999 年 5 月第一版 1999 年 5 月第一次印刷

印 数 / 1-5000 册

定 价 / 16.00 元 (册)

JS438/11

前言

Java 2 的许可证形式已经具有朝开源软件 (Open Source) 方向发展的趋势。记得 1998 年 12 月 8 日, 也就是在 Java 2 新的许可证发布的第二天, IT 工业界立即对 Sun Microsystems 公司的这一举措发表了热烈的评论。

Tim O'Reilly 先生 (您应该知道他是谁) 说:

“开源软件最强大的东西之一在于, 它一般是基于一个非常狭小的领域发展起来的, 某个人想解决对他非常重要的某个问题, 而这个问题刚开始时对于其他任何人来说看起来无关紧要, 但最终成了某种影响全局的东西。像 Java 这样的新技术, 如果它的用户群体能用它来解决某些特殊问题, 那么它的应用领域就会得到扩展, 甚至会出现革新。

“Java 是将来计算机工业的关键技术, 它支持网络化的智能设备。现在 Java 朝开源软件领域迈开了步伐, 因此对未来的美好憧憬会更快地变成现实, 而且它会带给我们更多有趣的惊喜。”

毫无疑问,在 Java 的发展史上,Java 2 的确是一个重大的里程碑。Java 的重要意义体现于:在网络化的计算机世界里,Java 实现了系统理论中“精缩影”与“广扩形”原理,在技术层次上取得了巨大的突破(注)。

对于中国的计算机工业界来说,Java 2 的确提供了一个千载难逢的良机。虽然在中国市场上尚未见到一个真正成熟的用 Java 开发的应用程序,但是我相信这样的程序很快就会面世。Java 一开始就支持 16 位的 Unicode 编码字符集,因此在某种程度上避开了软件开发项目中复杂的国产化和国际化的难题,起码开发纯中文的应用程序应该不会像现在这样别扭了。

然而,Sun Microsystems 公司现在提供的 Java 技术规范的原始文档长达厚厚的几千页,纵然您有一目十行之特异功能,从头到尾读完至少也要好几个月的时间,而且大部分人在读完原始开发文档之后,仍然觉得是雾里看花、难得要领。无怪乎有人曾说,Java 2 的发布实际上等于判处了业余 Java 程序员的“死刑”。

能否找到一种便捷的办法,让有志于成为 Java 程序员的初学者快速地领会 Java 编程原理呢?我们阅读了 O'Reilly 公司出版的 Java 系列英文原著后找到了肯定的答案,我们最终决定推出一套针对中国读者特点的 Java 系列著作来达到这一目的。

注:对系统理论有兴趣的读者可以参考吴学谋教授所著的《泛系导论》。这是一本熔哲理、数理和技术于一炉的著作。

在 O'Reilly 出版的 Java 系列著作中，《Java in a Nutshell》和《Java Power Reference》在美国市场上是极为成功的两本，其中每一章都写得非常精彩，每本著作都宛如一串钻石组成的项链。

当然，这两本著作都是针对中高级程序员创作的，初学者一口气吃下去难免消化不良。为了让初级 Java 程序员也能欣赏这些“阳春白雪”，我开始动手对原著进行一些不伤脾胃的改动。作为我们 Java 系列丛书中的第一本，我将《Java in a Nutshell》的第一章和《Java Power Reference》的第二章分别从原著中分离出来，再合并成这本小册子，专门介绍 Java 语言和 Java 平台。我的初衷是想让您借此获得一个关于 Java 语言和最新 Java 平台的总体印象。

虽然由于篇幅所限，这本小册子没有就每一个 Java 技术子专题深入展开讨论，但是，我又不想让您觉得这本书是那种不痛不痒的泛泛之作，而是试图将 Java 语言和最新 Java 平台的精髓向您和盘托出，让您从总体上把握 Java 的要领，在今后去阅读我们的后续著作或者 Java 的原始文档时，可以把这本小册子作为您 Java 探险之旅全程的指南针和寻宝路线图。

致谢

首先感谢《Java in a Nutshell》和《Java Power Reference》的原作者大卫·弗拉纳甘先生(David Flanagan)，他欣然同意我们根据中国国情将他的两本著作中的精彩片段分离出来，并组合成本书，更为可贵的是，他对如何分离原著和重新组织材料提供了详细指导。

感谢 O'Reilly 公司亚太地区行政总管唐明仁 (Ron Tomich) 先生, 他帮助我说服了原著作的各位编辑同意我对原材料的结构进行如此大胆的改变和重组, 这本小册子篇幅很短, 我们的尝试是否会在市场上收到良好的预期效果, 当时是不得而知的, 决策时具有很大的风险, 唐先生的这种远大眼光和敢于承担风险的勇气令我敬佩。

感谢 O'Reilly 公司的 Hanna Dyer 和 Edie Freedman, 她们充分理解了我们选题的良苦用心, 并为这本小册子设计了简单明快的封面。感谢凯瑟琳·肯妮 (Kathleen Kenny), 她为本书提供了插图和封面设计的图形文件。

在翻译这样前沿性的技术著作时, 我遇到的最大问题是怎样准确地把握大量新出现的技术术语的内涵和外延, 并找到最匹配的中文表达方式。倪文龙同志在翻译第二章时提供了一些协助, 在此表示感谢。但毫无疑问, 翻译中的所有错误均应该由我来负责。这本小册子在这方面究竟做得妥当与否, 还请广大读者来积极评判。

洪峰

奥莱理软件(北京)有限公司

1999年3月

JLangPlat@mail.oreilly.com.cn

作者简介

David Flanagan 是一位职业计算机程序员、用户界面设计师、教员。他在 O'Reilly 出版的作品包括《X Toolkit Intrinsic 参考手册》和关于 Motif 工具的著作：《流线型 GUI 设计》和《用 Xmt 库编程》。David Flanagan 先生获有 MIT 计算机科学学位。

译者简介

洪峰先生是计算机程序员、专业咨询作家，中国大陆推广应用 Linux 和开源软件的先锋之一。在到 O'Reilly & Associates 公司工作之前，他喜欢研究数学和哲学问题，而且是泛系理论的热心追随者。

目录

| | |
|----------|-----|
| 前言 | v |
| 致谢 | vii |

第一章 Java 语言

| | |
|----------------------|----|
| 1.1 Java 简史 | 1 |
| 1.2 Java 语言的特性 | 3 |
| 简单性 | 3 |
| 面向对象 | 4 |
| 分布式的特点 | 5 |
| 解释性的语言 | 6 |
| 健壮性 | 7 |
| 安全性 | 8 |
| 体系中立性 | 8 |
| 可移植性 | 9 |
| 高性能 | 9 |
| 多线程性 | 11 |
| 动态性 | 12 |

| | |
|---------------------|----|
| 1.3 简单的例子 | 12 |
| Hello World! | 12 |
| 一个 Java 小应用程序 | 13 |
| 浏览小应用程序 | 17 |

第二章 Java 平台

| | |
|-------------------------------|----|
| 2.1 Java 是什么? | 19 |
| Java 编程语言 | 20 |
| Java 平台 | 20 |
| 2.2 使用 Java 的关键利益 | 22 |
| 一旦写成, 随处运行 | 22 |
| 安全性 | 23 |
| 以网络为中心的编程 | 24 |
| 动态的、可扩展的程序 | 24 |
| 国际化 | 24 |
| 性能 | 25 |
| 程序员的效率和开发速度 | 26 |
| 2.3 Java 2 还是 Java 1.2? | 26 |
| 2.4 Java 2 的核心特征 | 27 |
| 语言基础 | 27 |
| 输入/输出 (I/O) | 28 |
| 联网 | 30 |
| 数学包 | 31 |
| 数据结构和工具 | 31 |
| 国际化 | 32 |
| 安全性 | 32 |

| | |
|----------------------------|----|
| 2.5 用户界面特性 | 33 |
| Swing 元件 | 34 |
| 模型 / 视图的分离 | 41 |
| 可插入的“观-感” | 41 |
| 可访问性 (Accessibility) | 43 |
| 各种 Swing 特性 | 43 |
| 拖-放 (Drag-and-Drop) | 45 |
| 输入方法支持 | 46 |
| 帮助 (JavaHelp) | 46 |
| 小应用程序 (Applet) | 47 |
| 豆容器 (JavaBeans) | 47 |
| 2.6 图形特性 | 49 |
| Java 2D | 50 |
| Java 3D | 53 |
| Java 多媒体框架 | 55 |
| 声音 | 55 |
| 2.7 企业特性 | 56 |
| 数据库访问 | 56 |
| RMI 分布式对象 | 57 |
| CORBA 的分布式对象 | 58 |
| 域名和目录 | 59 |
| 电子邮件 | 59 |
| 企业 JavaBeans | 60 |
| 小伺服器 | 61 |
| 2.8 还有更多的东西! | 62 |

本章提要

- ◆ Java 简史
- ◆ Java 语言的特性
- ◆ 简单的例子

第一章

Java 语言

1.1 Java 简史

Java 首先是作为家用电器，如电烤箱、微波炉和个人数字式助理等家电设备的编程语言面世的。家用电器的软件有独特的设计要求。例如，软件需要能在新型的芯片上工作得与原来的芯片一样好——由于新芯片的功能更强，或者具备新的特性，制造商经常更换所使用的芯片。对软件的要求是必须高度可靠，因为一般来说，当家用电器不能工作时，制造商不得不为用户更换全部设备。

Sun 公司一个在戈斯林（James Gosling）领导下的小组很快发现，现有的编程语言，如 C 或者 C++ 都不能完全满足这种要求。用 C 或者 C++ 编写的程序必须要针对某个特定的芯片编译后才能使用。当出现新的芯片后，原程序必须重新编译。同时，C 和 C++ 的复杂性使得编写可靠程序的难度很高。

在得出这一研究结论后，戈斯林在 1990 年开始动手设计更适合消费类家用电器软件要求的新型编程语言。这一语言刚开始时代号为“Oak”，它的规模很小、很可靠，而且与硬件体系无关。1993 年，在 Java 工作组继续设计这一新语言时，Internet 上出现了 WWW，而且发展势头如暴风骤雨一般猛烈，Java 的开发人员发现像 Java 这样与硬件体系无关的语言用于 Internet 编程是十分理想的，因为这样的程序可以运行在由不同类型的计算机连接而成的 Internet 网络上。事实上，Java 所有的设计目标使之非常适合于 Internet 编程。

这是 Java 的开发对于 Sun 来说所具有新的重要意义。原小组设计了一种 Web 浏览器，称为 HotJava，这是第一种支持 Java applet 的 Web 浏览器。所谓 applet 就是一个小的 Java 程序，它可以嵌入到另一个应用程序中去。你可以在 HTML 文档中嵌入 applet，从而在 Web 页面中提供交互式的、可执行的内容。HotJava 以一种非常直观的方式向程序员和其他人们展示了 Java 的威力，接着 Java 就飞黄腾达了。

在 Sun 不断升级 Java 版本时，程序员们开始创造各种漂亮的小应用程序。他们的经验和反馈信息对进一步完善这门语言及其“应用程序编程接口”(API) 提供了极有价值的帮助。这一切都使计算机界对它产生了更浓厚的兴趣。

1997 年初，Sun 推出了 Java 1.1 版本，在运行速度等性能上有了大幅度的提高，同时规模也在膨胀。在 1998 年下半年，Sun 推出了 Java 2 Platform，这已经是一个健壮而成熟的计算平台了。

1.2 Java 语言的特性

在 Sun 公司原来出版的技术白皮书中，Java 的定义如下：

Java：一种简单的、面向对象的、分布式的、解释型的、健壮的、安全的、体系中立的、可移植的、高性能的、多线程的、动态的语言。

Sun 公司承认这个定义的确很长，而且念起来很拗口，但实际上这个定义充分地描述了 Java 语言的性质。为了解 Java 的趣味性，让我们来考察这一定义中的每一个定语所描述的语言特征，看看 Sun 在设计这一语言时试图达到什么目的。

1.2.1 简单性

Java 是一种简单的语言。简单性究竟意味着什么呢？Java 的原设计小组的第一个目标是创造出一种程序员可以迅速学会的语言，因此这种语言的结构应该保留得很小。另一个设计目标是让这门语言在外观上应该让大多数程序员感到很熟悉，从而使程序员很容易掉头使用这一新的开发工具。如果你是 C 或者 C++ 程序员，那么你会发现 Java 使用了许多与 C 和 C++ 相同的结构。

为了让这一语言既小，又让人面熟，Java 的设计师们去掉了 C 和 C++ 中的许多特性，这些特性已经被以往的编程实践证明功能很差，或者运用得很少。例如，Java 不支持 goto 语句，取而代之的是 break 和 continue 语句，以及例外处理。Java 不使用头文件，并抛弃了 C 和 C++ 预处理程序。因为 Java 是面向对象的，所以 C 语言中的 struct 和 union 等结构也取消了。Java 还抛弃了 C++ 中的运算符重载和多继承等特性。

但是，也许重要的简化是 Java 没有使用指针。指针是在 C 和 C++ 编程实践中最容易引起出错的原因之一。由于 Java 不使用结构，而数组和字符串都是对象，因此没有使用指针的必要。Java 自动地为你完成参照通知和参照解除的过程。Java 还可以自动地完成“垃圾收集”的工作，你不用担心内存管理之类的事情。这一切把你从对指针游动和非法指针传递的担心中解脱出来，因此你可以将精力集中在开发程序的实际功能上。

如果你听说 Java 是一种从 C 和 C++ 中提取了精华而形成的语言时，你完全可以作出自己的判断。我认为 Java 的确是一门功能全面而且非常优美的语言。

1.2.2 面向对象

Java 是一种“面向对象”的语言。对于程序员这意味着，你可以将精力集中在应用程序的数据和对数据操作方法上，而不是严格按照过程的观点去思考。如果你熟悉 C 语言中基于过程的编程技术，那么在使用 Java 时，就需要改变设计思路。但是，一旦你熟悉了这一新编程原理的威力时，你就会迷上它。

在面向对象的系统中，“类”（class）是数据和运算这些数据的方法的集合体。数据和方法结合在一起后对一个“对象”（object）的状态和行为进行描述。类是按照某种层次进行安排的，所以子类可以继承其父类的行为。

类的层次之中有一个“根类”（root class），根类是一个具有非常一般性行为的类。

java.lang 包中的 object 类 (即 java.lang.object) 是所有 Java 类层次中的根类。

我们可以举一个自然界中的例子。你可以将“生物”定义为你类层次中的根类。该类可有两个基本的子类(subclass): “动物”和“植物”。你还可以将“哺乳动物”定义为“动物”的一个子类,最后,你还可以将“人”定义为“哺乳动物”的一个子类,“人”这个类将从“哺乳动物”类、“动物”类和“生物”根类中继承行为。

Java 带有一套扩展的类,它们被安排在“包”(package)中,你可以在程序中使用这些包。例如,Java 提供创建图形用户界面的类(在 java.awt 包中)、处理输入和输出的类(在 java.io 包中)、支持网络功能的类(在 java.net 包中),等等。

与 C++ 不同的是,Java 一开始就是为面向对象而设计的。Java 中的绝大多数东西都是对象,简单的数字、字符和布尔类型的数据是仅有的例外。当然,Java 还是设计得与 C++ 很相似,你会发现 Java 去掉了 C++ 中的许多复杂的东西。如果你是 C++ 程序员,那么你必须仔细地学习 Java 中面向对象的结构,尽管 Java 的语法与 C++ 很相似,但是这两者的行为却不那么接近。

1.2.3 分布式的特点

Java 支持基于网络的应用程序,它是一种分布式的语言。你可通过 java.net 包中的类来支持各种不同类型的网络连接。例如,URL 类允许 Java 应用程序打开并存取分布于 Internet 上的远程对象。

有了 Java，打开远程的文件就跟打开本地的文件一样容易。远程方法调用应用程序编程接口（RMI API，即 Remote Method Invocation API）可以让调用远程对象的方法就像调用本地对象的方法一样。

Java 还可以通过 Socket 类支持传统的低级别的网络连接，例如，基于数据报（datagram）和基于流（stream-based）的网络连接。这样，你可以创建分布式的客户机（client）和服务程序（server）程序。

当 Java 分布式的特点真正与动态地装载类的能力结合在一起时，Java 便会大放光彩。这些特性使 Java 的解释器跨网络地下载并运行代码成为可能。例如，Web 浏览器可以从 World Wide Web 下载并运行一个 Java 小应用程序（applet）就是按照这一机制运作的。我们下面将看到，Java 具有健壮的安全性机制可以保证这一功能得以安全地实现。但是，具体方案可以比这更加复杂。我们可以假设一个用 Java 编写的多媒体字处理器，当要求程序显示从来没有遇到过的数据类型时，这个字处理器可以动态地从网络上下载一个类来解析该数据，然后再动态地下载另一个类（或许是一个 Java 豆容器）来在一个混合的文档中显示该数据。像这样使用分布在网络上的资源时，应用程序可以动态地扩大功能，并适应用户要求的变化。

1.2.4 解释性的语言

Java 语言编译器产生“字节码”（bytecode），而不是原始的机器码。在实际运行 Java 时，你要利用解释器来执行编译过的字节码，所以 Java 是一门解释性的语言。Java 的字节码提供了一种与硬件体系无关的对象文件格式，该代码可以多