

Java 自 1996 年正式发布以来,经历了初生、成长和壮大阶段,现在已经成为 IT 领域里的主流编程语言。Java 起源于 Sun 公司的一个名为“Green”的项目,目的是开发嵌入家用电器的分布式软件系统,使电器更加智能化。图 P-1 为参与 Green 项目的开发人员。Green 项目一开始准备采用 C++ 语言,但是考虑到 C++ 语言太复杂,而且安全性差,于是决定基于 C++ 语言开发一种新的 Oak 语言(即 Java 的前身)。



图 P-1 参与 Green 项目的开发人员

Oak 是一种适用于网络编程的精巧而安全的语言,它保留了许多 C++ 语言的语法,但去除了明确的资源引用、指针算法与操作符重载等潜在的危险特性。并且 Oak 语言具有与硬件无关的特性,制造商只需更改芯片,就可以将烤面包机上的程序代码移植到微波炉或其他电器上,而不必改变软件,这就大大降低了开发成本。

当 Oak 语言成熟时,Internet 也在全球迅速发展。Sun 公司的开发小组认识到 Oak 非常适合 Internet 编程。1994 年,他们完成了一个用 Oak 语言编写的早期的 Web 浏览器,称为 WebRunner,后改名为 HotJava,展示了 Oak 作为 Internet 开发工具的能力。

1995 年,Oak 语言更名为 Java 语言(以下简称为 Java)。Java 的取名有一个趣闻:据说有一天,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,并得到了其他人的赞赏。于是,Java 这个名字就这样传开了。

1996 年,Sun 公司发布 JDK1.0,计算机产业的各大公司(包括 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracel、Toshiba 和 Microsoft 等)相继从 Sun 公司购买了 Java 技术许可证,开发相应的产品。

1998 年,Sun 公司发布了 JDK1.2(从这个版本开始的 Java 技术都称为 Java2)。Java2 不仅兼容于智能卡和小型消费类设备,还兼容于大型服务器系统,它使软件开发商、服务提供商和设备制造商更加容易抢占市场。这一开发工具极大地简化了编程人员编制企业级 Web 应用的工作,把“一次编程,到处使用”的诺言应用到服务器领域。

1999 年,Sun 公司把 Java2 技术分成 J2SE、J2EE 和 J2ME。其中 J2SE 就是指从 1.2 版本开始的 JDK,它为创建和运行 Java 程序提供了最基本的环境。J2EE 和 J2ME 建立在 J2SE 的基础上,J2EE 为分布式的企业应用提供开发和运行环境,J2ME 为嵌入式应用(比如运行在手机里的 Java 程序)提供开发和运行环境。

Java 的公用规范(Publicly Available Specification, PAS)在 1997 年被国际标准化组织(ISO)认定,这是 ISO 第一次破例接受一个具有商业色彩的公司作为公用规范 PAS 的提交者。

Foreword

2000年和2002年，Sun公司分别发布了JDK1.3和JDK1.4，它们在原先版本的基础上做了一些改进，扩展了标准雷库，提高了系统性能，还修正了一些Bug。

2004年，Sun公司发布了JDK5.0，这一版本做了重大的改进，增加了泛型机制、自动装箱/拆箱和foreach语句等。这一版本原称为JDK1.5，在2004年的SunOne会议后，更名为JDK5.0。

2006年，Sun公司发布了JDK6，该版本没有在语言方面进行大的改进，主要是改进了运行性能，并增强了类库功能。

随着以操纵海量数据为主的软件系统越来越依赖商业硬件，而不是专用服务器，以Solaris操作系统为主打产品的Sun公司变得越来越不景气，最终于2009年被Oracle公司收购。Java的发展停滞了很长一段时间。直到2011年，Oracle公司发布了JDK7，做了一些简单的改进。

2014年，Oracle公司发布了JDK8，这一版本也做出了重大的改进，其中比较显著的改进包括：

- l 引入能够简化编程的Lambda表达式。
- l 接口中允许包含提供了实现的默认方法和静态方法。
- l 引入功能强大的Stream API。
- l 引入方便实用的处理日期和时间的新的Date/Time API。
- l 引入避免空指针异常的Optional类。

总之，面向对象的Java语言具备“一次编程，任何地方均可运行”的能力，使其成为服务提供商和系统集成商用以支持多种操作系统和硬件平台的首选解决方案。Java作为软件开发的一种革命性的技术，其地位已被确定。如今，Java技术已被列为当今世界信息技术的主流之一。表P-1对Java的发展历史做了总结。

表 P-1 Java 发展历史

年份	Java 发展历史
1995	Java 语言诞生
1996	JDK1.0 发布，10 个最主要的操作系统供应商申明将在其产品中支持 Java 技术
1997	JDK1.1 发布
1998	JDK1.1 下载量超过 200 万次，JDK1.2（称 Java2）发布，JFC/Swing 技术发布，JFC/Swing 被下载了 50 多万次
1999	Java 被分成 J2SE、J2EE 和 J2ME，JSP/Servlet 技术诞生
2000	JDK1.3 发布，JDK1.4 发布
2001	Nokia 公司宣布到 2003 年将出售 1 亿部支持 Java 的手机，J2EE1.3 发布
2002	JDK1.4 发布，自此 Java 的计算能力有了大幅度提升。J2EE SDK 的下载量达到 200 万次
2003	5.5 亿台桌面上运行 Java 程序，75% 的开发人员将 Java 作为首要开发工具
2004	JDK1.5 发布，这是 Java 语言发展史上的又一里程碑事件。为了表示这个版本的重要性，JDK1.5 更名为 JDK5.0
2005	JavaOne 大会召开，Java 的各种版本被更名，取消其中的数字“2”：J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME
2006	JDK6 发布
2009	Sun 公司被 Oracle 公司收购
2011	JDK7 发布
2014	JDK8 发布，对 Java 语言特性做了重大改进，增加了 Lambda 表达式

Java 语言的特点

Java 应用如此广泛是因为 Java 具有多方面的优势。其特点如下：

(1) 面向对象。Java 自诞生之时就被设计成面向对象的语言，而 C++ 语言是一种强制面向对象的语言。面向对象可以说是 Java 最重要的特性，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重用。

(2) 跨平台。对于 Java 程序，不管是 Windows 平台还是 UNIX 平台或是其他平台，它都适用。Java 编辑器把 Java 源程序编译成与体系结构无关的字节码指令，只要安装了 Java 运行系统，Java 程序就可在任意的处理器上运行。这些字节码指令由 Java 虚拟机来执行，Java 虚拟机的解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

(3) 直接支持分布式的网络应用。除了支持基本的语言功能，Java 核心类库还包括一个支持 HTTP、SMTP 和 FTP 等基于 TCP/IP 协议的类库。因此，Java 应用程序可凭借 URL 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。在 Java 出现以前，为分布式环境尤其是 Internet 提供动态的内容无疑是一项非常宏伟、难以想象的任务，但 Java 的语言特性却使我们很容易地达到了这个目标。

(4) 安全性和健壮性。Java 致力于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 支持自动内存管理，这不但让程序员减轻了许多负担，也减少了程序员犯错的机会。Java 自己操纵内存减少了内存出错的可能性。Java 还能够检测数组边界，避免了覆盖数据的可能。在 Java 语言里，指针和释放内存等功能均被抛弃，从而避免了非法内存操作的危险。

以上特点，是 C++ 语言及其他语言无法比拟的（C++ 语言尽管也是面向对象的，但并不是严格意义上的面向对象的语言）。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。Java 发展到现在，已经不仅仅是一种语言，可以说是一种技术，这个技术涉及网络和编程等领域。另外，Java 是非常简单、高效的，有调查数据发现：用 C++ 和 Java 来做一个相同功能的项目，用 Java 写的程序要比用 C++ 写的程序节省 60% 的代码和 66% 的时间。可以说，用 Java 语言编程时间短、功能强，编程人员接手起来更容易、更简便。

本书的组织结构和主要内容

本书以 6 条主线贯穿全书：面向对象编程思想、Java 语言的语法、Java 虚拟机执行 Java 程序的原理、在实际项目中的运用、设计模式和性能优化技巧。书的每一章都会围绕若干条主线来展开内容，并且根据全书的布局，合理安排每一章内容的深度。本书主要内容包括：面向对象开发方法概述、第一个 Java 应用、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、类的生命周期、对象的生命周期、内部类、多线程、数组、Java 集合、泛型、Lambda 表达式、输入与输出 (I/O)、图形用户界面、常用 Swing 组件、Java 常用类和 Annotation 注解。

这本书是否适合您

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。常常会有读者问我这样的问题：

我学了 Java 已经一年多了，现在就只能用 JSP 写点东西，其他的东西实在太多了，我整天学都学不完，很迷茫，不知道该如何有针对性地去学，以找到一份 Java 工作，现在是困死在 Java 里了。

撰写本书，目的之一是为了帮助读者看清 Java 万花筒的本质，从复杂的表象中寻找普遍的规律，深刻理解 Java 的核心思想，只有掌握了普遍的规律与核心思想，才能以不变应万变，轻轻松松地把握 Java 技术发展的新趋势，迅速地领略并且会熟练运用一门新的技术，而不成为被动的追随者，知其然而不知其所以然。

阅读本书，读者对 Java 的领悟将逐步达到以下境界：

- | 熟悉 Java 语法，熟练地编译和调试程序。
- | 按照面向对象的思想来快速理解 JDK 类库，以及其他第三方提供的类库，通过阅读 JavaDoc 和相关文档，知道如何正确地使用这些类库。
- | 按照面向对象的思想来分析问题领域，设计对象模型。
- | 在开发过程中会运用现有的一些优秀设计模式，提高开发效率。
- | 当一个方法有多种实现方式时，能够从可维护、可重用及性能优化的角度选择最佳的实现方式。
- | 理解 Java 虚拟机执行 Java 程序的原理，从而更深入地理解 Java 语言的各种特性和语法规则。

本书的技术支持网站

本书的技术支持网站为：www.javathinker.net。读者可以在该网站交流 Java 技术，提出本书的勘误信息，作者会在本网站为读者答疑。本书中实例源文件及思考题答案的下载网址为：www.javathinker.net/download.htm。

致谢

本书在编写过程中得到了 Oracle 公司在技术上的大力支持，飞思科技产品研发中心负责编辑审核。此外，复旦软件学院的戴开宇等老师为本书的编写提供了有益的帮助，JavaThinker.net 网站的网友们为本书的升级提供了许多宝贵建议，在此表示衷心的感谢！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。

